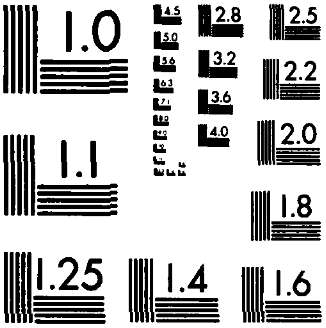
AD-A155 081   FURTHER EVALUATION OF COMPUTER-AIDED MESSAGE HANDLING   1/1
(U) MITRE CORP BEDFORD MA N C GOODWIN ET AL. 30 SEP 80
MTR-8149 F19628-80-C-0001

UNCLASSIFIED   F/G 9/2   NL


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

1

MITRE-Bedford Division

MTR-8149

Further Evaluation of Computer-Aided Message Handling

N. C. Goodwin
S. W. Hosmer
D. G. Miller

30 September 1980

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION IS UNLIMITED (A)

"Review of this material does not imply Department of Defense endorsement of factual accuracy or opinion."

Approved for public release; distribution unlimited.

MITRE

MITRE Technical Report

MTR-8149

Further Evaluation of Computer-Aided Message Handling

N. C. Goodwin
S. W. Hosmer
D. G. Miller

30 September 1980

CONTRACT SPONSOR	DARPA
CONTRACT NO.	F19628-80-C-0001
PROJECT NO.	8070
DEPT.	D73

"Review of this material does not imply Department of Defense endorsement of factual accuracy or opinion."

Approved for public release; distribution unlimited.

THE MITRE CORPORATION
BEDFORD, MASSACHUSETTS

Department Approval: J.A. Clapp
J. A. Clapp

MITRE Project Approval: David G. Miller
David G. Miller

ABSTRACT

The Military Message Experiment was an operational evaluation of a computer-aided message handling system, used by members of CINCPAC's Operations Directorate for various message handling tasks. The system enabled messages to be delivered to action officers faster and with less effort than they were with manual procedures, facilitating access to a broad range of infotmation and giving its users more control over the messages they read. Lessons learned from the experiment should influence the design of future automated message handling systems. This report describes a follow-on effort, designed to supplement user-oriented data drawn from the experiment and presented in previous reports, and is the last one planned in the series.

See p 1

TABLE OF CONTENTS

Section		Page
LIST OF ILLUSTRATIONS		ix
LIST OF TABLES		x
1	INTRODUCTION	1
	THE EXPERIMENT	1
	THE FOLLOW-ON EFFORT	2
2	INSTRUCTION ENTRY PREFERENCES	4
	INTRODUCTION	4
	DISCUSSION	4
	CONCLUSIONS	7
3	INSTRUCTIONS AND ERRORS	8
	DISCUSSION	8
4	PATTERNS OF INSTRUCTION USE	11
	DISPLAY AND VIEW	11
	DELETE	13
	GET AND KEYWORD	14
	DISCUSSION	17

TABLE OF CONTENTS (Continued)

Section		Page
5	SELECTION CRITERIA	19
	INTRODUCTION	19
	CRITERIA USE	19
	NAMED VS. UNNAMED SELECTORS	22
	USER TYPES AND CRITERIA USE	22
	CONCLUSIONS	25
6	SELECTION LEVEL	26
	DISCUSSION	26
7	THE "RESTORE" INSTRUCTION	28
	INTRODUCTION	28
	DISCUSSION	28
	CONCLUSIONS	30
8	RETRIEVAL OF MESSAGES FROM THE ARCHIVE	31
	INTRODUCTION	31
	RETRIEVED VS. NON-RETRIEVED MESSAGES	31
	The Data Tables	31
	The Data	32
	AUTODIN VS. IN-PREP MESSAGES	36
	CONCLUSIONS	36

TABLE OF CONTENTS (Continued)

Section		Page
9	READBOARD USAGE	39
	INTRODUCTION	39
	USE OF READBOARDS	39
	CONCLUSIONS	41
10	TRACKING USERS	43
	INTRODUCTION	43
	USER A	43
	USER B	47
	USER C	50
	USER D	53
	USER E	56
	USER F	56
	CONCLUSIONS	61
11	DATA ACQUISITION METHODS	63
	INTRODUCTION	63
	SESSION TRANSCRIPTS	63
	The Form of Session Transcripts	64
	Preprocessing the Session Transcripts	65
	The Preprocessing Programs	67

TABLE OF CONTENTS (Continued)

Section	Page
SESSION TRANSCRIPT DATA PROCESSING PROGRAMS	67
Raw Instruction Counting Program	67
Typed Instruction Counting Program	68
Error Correlation Program	68
Pattern Counting Program	69
Selector Counting Program	70
Restrict and Augment Level Counting	70
NON-SESSION TRANSCRIPT DATA PROCESSING PROGRAMS	71
Archive Retrieval Time Information Program	71
Data Collection Facility Programs	71
LIST OF REFERENCES	73
APPENDIX A RAW SESSION TRANSCRIPT DATA COUNTS	75
GLOSSARY	81
DISTRIBUTION LIST	83

LIST OF ILLUSTRATIONS

Figure Page

1 Age of Requested Messages 35

2 Retrieval of In-Prep and Incoming Messages 37

3 System Up Time 44

4 User A: Sigma Use 45

5 User B: Sigma Use 48

6 User C: Sigma Use 51

7 User D: Sigma Use 54

8 User E: Sigma Use 57

9 User F: Sigma Use 59

10 Sample Session Transcript 65

LIST OF TABLES

Table		Page
1	Use of Function Keys vs. Typed Instructions	5
2	Instruction Entry Preferences	6
3	Instructions and Errors	9
4	Display/View Instructions	12
5	Delete Instructions	15
6	Get and Keyword Instructions	16
7	Unnamed Selectors	21
8	Named (Saved) Selectors	23
9	Selection Level Occurrence	26
10	Items Created, Deleted, and Restored	29
11	Number of Immediate Restores	30
12	Message Retrieval: All Messages	33
13	Message Retrieval: In-Prep and Incoming Messages	34
14	Readboard Use	40
15	User A: Instruction Use	46
16	User B: Instruction Use	49
17	User C: Instruction Use	52
18	User D: Instruction Use	55

LIST OF TABLES (Concluded)

Table		Page
19	User E: Instruction Use	58
20	User F: Instruction Use	60
A-1	Frequently Used Function Key Instructions	76
A-2	Frequently Occurring Error Messages	77
A-3	First Words in Typed Instructions	78
A-4	Second Words in Typed Instructions	79

SECTION 1

INTRODUCTION

THE EXPERIMENT

The Military Message Experiment was an operational evaluation of the utility of computer-aided message handling in a military environment. The experiment was conducted at the headquarters of the Commander-in-Chief Pacific (CINCPAC) from May 1977 to September 1979. The headquarters are located at Camp Smith, Hawaii; the Operations Directorate (J3) at CINCPAC was chosen to participate in the experiment because of their high message handling activity.

The Sigma message system, developed by the Information Sciences Institute of the University of Southern California, was selected for use in the experiment. Sigma provided users with automated message handling functions for message distribution, message readdressal, incoming message review, file maintenance, selective retrieval of messages, retrieval of messages from archive, creation, coordination and release of formal AUTODIN messages, and creation of informal internal memos and notes.

Sigma was installed at CINCPAC in May, 1977. In October, 1978, limited experimental use (LEU) of the system began but system improvements continued. In February, 1979, the period of full experimental use (FEU) began; users were directed to use the system as their primary message handling system although the manual procedures were still maintained. In addition, for three weeks in March, 1979, an exercise simulating crisis situations was conducted at CINCPAC. Sigma was used for message handling activities during the exercise.

The objectives of the experiment were three-fold. The first objective was to evaluate the utility of automated message handling in a military environment. Sigma's functionality and the utility of its features to CINCPAC personnel were examined. A second objective was to evaluate the user interface of the system (Sigma) as an aid to specifying the design of future automated message systems. The third objective was to evaluate the organizational impact of the automated system; that is, the ways in which users' roles and activities changed with computer-aided message handling.

In support of these objectives, four types of data were collected. The system itself collected data with an on-line data collection facility (DCF). The DCF recorded information about the

Table 5

Delete Instructions

Instruction Number	 % Of All	Followed By	% Of Total For Instruction
delete entry (list)		finish displayed object	16%
		delete entry	16%
8,537	2%	backup one	15%
		display file	13%
		delete file entry	7%
delete file entry		delete file entry	60%
		delete next entry	9%
37,859	11%	move entry	4%
		view next entry	4%
		display file entry	4%
delete next entry		delete next entry	78%
		move entry	6%
37,226	11%	view next entry	4%
		delete file entry	3%
		display next entry	2%

instruction, "delete entry", was also often followed by another delete instruction; closing ("finishing") the open file was another frequent action. These pairings indicate that users often spent time removing unwanted messages from a file, rather than mixing deletions with other actions. ("Move entry" moved the message into another file and deleted it from the present one.)

The presence of "backup one" as a frequent pairing with the multiple delete instruction indicates that users often did a selective retrieval ("restrict") to find a group of messages, deleted all or some of them, and then returned to the original set. "Restrict" was a useful capability for finding sets of messages; it will be discussed further in the following sections.

GET AND KEYWORD

The "get" instruction permitted a user to access information in another user's directory. "Keyword" was used to add a keyword to an entry or list of entries in a file. The keyword could then be used as a parameter for selective retrieval. Keywords were not displayed with entries, but a user could view the keywords associated with a file. Frequent pairings with these instructions are shown in Table 6.

"Get" was most often followed by an instruction that would display or view the object just obtained. (Selectors could not be put in the display window; they could only be viewed.) The next most frequent instruction after "get file" and "get text" was another "get". The ability to obtain multiple objects from another user would have had some value, but the relatively infrequent use of successive "gets" suggests that multiple "gets" should not be provided at the expense of system performance.

Users of "get" sometimes checked their directory to confirm that the object had, in fact, been successfully transferred. A message confirming that the "get" had been accomplished would have eliminated the need for this checking. A routine "your instruction has been processed" type of confirmation is not sufficient. The confirmation should include the object name and source (which have been entered by the user).

"Keyword" was often followed by another "keyword" instruction. This happened when a user was placing different keywords on various entries, but it also happened when users wanted to place several keywords on the same set of entries. Specification of multiple and response time. Reducing successive display or open file types of instructions would save both the system and the user from unnecessary effort.

"Display file entry" was used to display messages not necessarily in sequence in the file. The most common instruction following this was "show open file", which returned the file listing to the display window. The second most popular, "finish displayed object", had the same effect. The frequent use of "finish displayed object" indicates that some users did not realize the message being displayed would have been finished, or closed, when another message was displayed or opened. In any case, the high percentage of these two instructions following "display entry" indicates a need to return to the file display to decide on the next action.

"Display next entry" was most often followed by another execution of the same instruction. This highlights the importance of giving users the ability to cycle through a file. They should not have to return to the basic file display before displaying another message. However, the ability to return to the file display at will is important, as evidenced by the frequent use of the "show open file" instruction.

Messages placed in the view window were often checked prior to some action being taken. Thus the three versions of "view entry" ("view entry", "view file entry", "view next entry") were most often followed by instructions that deleted, moved, or routed the entry. Clearing the view window and viewing another entry were also frequent actions.

Clearing the view window had the effect of removing the contents of the view window from the display and expanding the display window to its full size. Thus a user might clear the view window prior to taking an action on a message, or might take action and then clear the view window.

DELETE

"Delete entry" was a typed instruction that could be used to delete a group of messages. "Delete file entry" and "delete next entry" were each function keys that deleted a single entry. The instructions that followed these delete instructions most frequently are shown in Table 5. It is interesting to note that the single entry delete instructions were followed a large percentage of the time by another delete instruction. The multiple entry delete

Table 4

Display/View Instructions

Instruction Number	% Of All	Followed By	% of Total For Instruction
display file		display file entry	17%
		display file	16%
24,942	7%	delete file	7%
		delete file entry	6%
		clear view window	5%
display file entry		show open file	45%
		finish displayed object	17%
18,836	5%	print displayed object	6%
		display next entry	5%
		print file entry	4%
display next entry		display next entry	45%
		show open file	28%
9,178	3%	delete file entry	8%
		finish displayed object	3%
		print displayed object	3%
view entry		delete entry	21%
		clear view window	20%
179	<1%	move entry	11%
		view entry	8%
		route	6%
view file entry		clear view window	26%
		view file entry	13%
8,644	2%	delete file entry	11%
		route	10%
		move entry	8%
view next entry		delete file entry	27%
		route	19%
10,879	3%	move entry	17%
		view next entry	12%
		clear view window	7%

SECTION 4

PATTERNS OF INSTRUCTION USE

Patterns of instruction use were examined for two purposes. One was to discover sets of instructions or functions that could be usefully combined into a single instruction; the other was to determine where the patterns suggest that users received inadequate feedback at the time of instruction execution.

Instruction pairings were recorded for the most commonly used functions. These were displaying a file or message (entry); viewing a message (entry); and deleting a file or message. (Although the "route" instruction was heavily used, it was designed for personnel with the specialized job of distributing messages. Since it was not used by most of the CINCPAC users, "route" was not included in this analysis.) Instruction pairings for getting files, selectors, and text, and for adding keywords to entries were also recorded.

Note that instruction pairs were counted for every occurrence of an instruction. Therefore, if an instruction were executed three times in succession, two pairs would be recorded.

DISPLAY AND VIEW

"Display" was used to put a file, message, or text into the display window where editing and manipulation were permitted. "View" placed a message, text, or selector into the view window, which was a read-only window. Pairings for these instructions are shown in Table 4.

The instruction that most often followed "display file" was "display file entry". However, a close second in frequency was another "display file" instruction. This suggests that the user did not display the proper file the first time, and so tried another file. The file directory showed a list of the files, their classifications, and owners. Some additional information about the file contents would have been helpful, so that a user would have had more confidence that the file being requested was, in fact, the correct one. The ability to associate a brief comment or note with the file entry in a directory or index listing might have been used. Although displaying the wrong file is not a major problem, opening a file for display is potentially costly in terms of system resources both the beginning and the end of the text being copied. Another frequent error message, "A pickup is illegal from the view window" (24%), occurred because users forgot that text could not be erased from the view window (which was intended only as a reference window). However, since Sigma copied the selected text anyway, there was no real penalty attached to using the pickup instruction incorrectly and many people continued to use it. The message "Text cannot be entered here" (22%) occurred most often when users attempted to copy information from the display and view windows into the instruction line. For future systems, this might be a useful capability.

30% of the user errors encountered when using the "readdress" instructions occurred because the message being readdressed was not an incoming AUTODIN message, but either a draft message or an internal memo or note. The rest of the "readdress" errors occurred when there was no indicated message to readdress.

The vast majority (82%) of the errors associated with "release message" occurred because users did not have authority to release outgoing messages. This authority was reserved for division and directorate management, and users lower in the hierarchy were expected to send messages upward for approval and release.

Note that generally the more frequently used instructions had smaller error rates; only 1% of the most frequently used instruction, "delete file/next entry", resulted in errors. As Table 3 shows, of the 10 instructions that were issued more than 10,000 times during the experiment, none resulted in more than 4% errors. This suggests that once users became familiar with Sigma and its basic functions, they made fewer errors with the instructions they used daily. The instructions which resulted in the most errors were less frequently used instructions.

Table 3

Instructions and Errors

		User Caused Errors	
Instruction	Number Issued	Number	Percent
copy message	139	113	81%
coordinate message	1095	233	21%
copy/move/pickup/put text	2820	481	17%
readdress message	723	104	14%
restore	929	130	14%
release message	1281	131	10%
display message (T)	996	40	4%
display text	5376	189	4%
clear view window	19165	673	4%
view message (T)	240	8	3%
finish	16640	434	3%
view directory	2971	75	3%
display message (F)	31685	758	2%
view message (F)	20801	381	2%
print	13570	222	2%
file/move message	20788	298	1%
delete message (F)	77833	974	1%
route	14128	134	1%
restrict/augment	19049	170	1%
empty/sort file	1297	8	1%
delete message (T)	8913	35	*
display file	39382	62	*
All Instructions	353031	7535	2%

* - <0.5%

SECTION 3

INSTRUCTIONS AND ERRORS

DISCUSSION

This discussion will relate specific instructions to the errors they caused. Instructions which caused the most difficulty will be identified and discussed. Table 3 shows, for several Sigma instructions, the total number of instructions issued during FEU and the number and percentage of errors each caused. It should be noted that only 7,535 error messages are included in this table, while 19,000 error messages are reported in Appendix A. This is because the remaining 11,465 session transcript entries recorded as errors could be more accurately described as system status messages. Only those messages attributable to user-created errors are discussed in this section.

The "copy message" instruction caused the greatest percentage of errors (81%). This instruction was designed to allow users to make a second copy (with a new message identification number) of a message being drafted for eventual release. However, incoming messages could not be copied; this would have created a potentially confusing situation in which comments made on one copy of an incoming message might not have been seen by those displaying another copy. About 67% of the errors caused by "copy message" were "You cannot copy incoming messages". Had that capability existed, only 13% of the "copy message" instructions would have resulted in errors, a figure more in line with the other instructions. Had the instruction been named "copy draft", it is likely the number of errors would have been less.

"Coordinate message" caused the second highest percentage of errors (21%). The most frequent of them (132 occurrences), "You have not specified any coordinators in the coordination list", occurred either when the drafter put no names in the coordination list, or when he forgot to highlight at least one of those names for the first coordination cycle. This reflects the complexity of the coordination process in general.

17% of the four instructions which manipulated text in text objects or messages, "copy text", "move text", "pickup text", and "put text", resulted in errors. One frequent error was "Two HEREs are needed for *copy text*/*move text*/*pickup text*" (29%). This occurred because many users forgot that it was necessary to mark

CONCLUSIONS

It was evident from the questionnaires that users preferred function keys to typing whenever they were available. However, 30% expressed a preference for the menu approach to instruction entry. With a menu, several options are displayed and the user marks the option he wants. This approach has several advantages. No typing skills are necessary, and usually only a single keystroke is needed. The system does not have to parse the entry, saving processing time. Menus may be useful in applications in which not all the user's options can fit on a usable function key layout.

In future designs, instruction entry must be made as convenient and efficient as possible. Both function keys and menus provide users with simple and fast approaches to entering and executing commands.

determine a message's entry number and type it in. (If a message was being displayed, the user would have to return to the file to determine the entry number, a further inconvenience.) In this case, the advantages of using function keys instead of typed instructions were not only in saving typing; there were also savings in repeated confirmations and system processing time required to return to the file display.

In most cases, function key use was over 95% compared to typed instruction use. Two exceptions to this were the "copy message" and "delete message" instructions. In the case of "copy message", the sample size was small (23 executions altogether), too small to draw any conclusions. In the case of "delete message", though, the sample size was 82,043, 10% of which were typed instructions. This relatively high figure for the typed version of the instruction was due to the fact that function key versions of "delete message" only deleted one message at a time, while the typed version could delete one or more messages at a time. When a user wished to delete several messages, it was more convenient to use the typed instruction to use a function key repeatedly.

On the final questionnaires, users were asked to select the technique they would prefer for entering frequently used items such as instructions. Table 2 presents the responses to that question. Fifty-two percent of the users preferred function keys. Users suggested that the more frequently used instructions such as "display file pending" should have been on function keys.

Table 2

Instruction Entry Preferences

	Responses	Percent of Total
Typing	6	18%
Function Keys	17	52%
Displayed Menu	10	30%
	33	100%

The number of executions and percent of total executions for both function keys and typed instructions are shown.

As Table 1 demonstrates, users preferred function keys for command entry overwhelmingly. The main reason for this was that it was quicker and easier to press a function key than to type an instruction. Function keys required only one keystroke; typing could require several, leaving more room for mistakes. Function key instructions required no confirmation by users, while all but advanced users had to confirm typed instructions before they were executed.

Table 1

Use of Function Keys vs. Typed Instructions

	Typed Instructions		Function Key	
	Number	Percent	Number	Percent
display message	847	3%	29296	97%
view message	149	1%	18763	99%
print	266	2%	12263	98%
delete message	8343	10%	73700	90%
reply message	0	0%	98	100%
copy message	2	9%	21	91%
readdress message	29	5%	564	95%

These data must be interpreted with some care. No instruction was exactly the same on a function key and as a typed instruction. For example, "display next entry" was a function key instruction, while "display entry <number>" was the corresponding typed instruction. When reviewing a series of entries in a message file, most users found it simpler to hit the function key than to

SECTION 2

INSTRUCTION ENTRY PREFERENCES

INTRODUCTION

This section covers Sigma users' preferences for function keys vs. typed instructions. For some instructions users had a choice of entry techniques; equivalent actions could be accomplished by typing an instruction or by pressing a function key. For some of these instructions the entry techniques were recorded and compared. These data were collected from Sigma's Data Collection Facility (DCF). In addition, comments about function keys vs. typed instructions in the questionnaires distributed at the end of the experiment are discussed.

Sigma supported two levels of user expertise for entering typed instructions. The "novice" user was required to hit two "executes" or "confirms" to execute an instruction. The "intermediate" user was only required to hit the execute key once. Users could change levels by contacting the experiment staff. Many CINCPAC users stayed at the novice level of entry. Some had their level changed to intermediate.

If Sigma could not understand a typed instruction, users had to edit the instruction and re-confirm. Typed instructions could be abbreviated; Sigma's command language processor (CLP) then expanded the instruction and either requested a confirm (novice users only) or immediately executed the expanded instruction.

Function keys, on the other hand, required only one keystroke and were already syntactically correct; they were not processed by the CLP and consequently response time was faster. Since the function keys were easier and faster to use, the system designers placed many of the instructions that they expected to be heavily used on function keys. Among these were, for example, "display file entry", "display next entry", "delete file/next entry", "view file entry", etc.

DISCUSSION

Table 1 presents data for instructions for which there was a choice of using a typed or function key version of an instruction.

use of the system by several users studied over the entire nine months of the experiment. Each of these areas will be discussed in depth and supporting data will be presented. Implications for future automated message handling systems will also be discussed.

The last section of this report presents, in detail, how many of the data for this report were produced.

instructions issued, the objects (messages, files, etc.) dealt with and their security level, the CPU time used, and the real-time Sigma's internal processing. The system also produced session transcripts, a human-readable record of the dialogue between each user and Sigma. Each syntactically correct instruction and its system response were recorded, along with the time at which they occurred. In addition, questionnaires were distributed to users, and interviews were conducted to determine how users felt an automated message system could help them accomplish their daily message handling tasks.

These data were collected at CINCPAC, and reduced and analyzed at MITRE. The results of the analyses, primarily of DCF and questionnaire data, were documented in Reference 1.

The experiment ended in September, 1979. Sigma was removed from CINCPAC and users returned to manual message handling procedures.

THE FOLLOW-ON EFFORT

While the data produced by the session transcripts were not as detailed as those produced by the DCF, they do contain valuable information not contained in the DCF, such as the arguments used in the "restrict" and "augment" instructions. The objective of the follow-on effort is to study areas of interest that were not studied in the earlier experiment evaluation. The session transcripts were not used extensively as a source of data for that evaluation but the additional information they provide is valuable. The session transcripts were the primary source of data for this follow-on effort.

Nine areas of Sigma use are discussed in this report: the use of function keys vs. typed instructions for command entry; errors correlated with the instructions that caused them; patterns of instruction use that occurred frequently; the criteria used for selecting a subset of messages from a file; the depth of selection necessary to reach a desired subset of messages; use of an instruction which restored deleted objects; the retrieval of messages from Sigma's archive; use of the on-line readboards; and

Table 6

Get and Keyword Instructions

Instruction		Followed By	% Of Total
Number	% Of All		For Instruction
get file		display file	44%
		get file	43%
268	<1%	view file directory	8%
get selector		view selector	83%
		clear view window	3%
78	<1%	view selector directory	3%
get text		display text	43%
		view text	25%
75	<1%	get text	10%
		view text directory	4%
keyword		keyword	25%
		view file entry	17%
94	<1%	restrict file	14%
		delete file entry	11%

keywords with a single instruction would be useful. Of course, showing the keywords associated with entries is also desirable.

The other instructions following "keyword" were generally popular in their own right; their pairings with "keyword" may only signify that popularity.

DISCUSSION

The investigation into instruction pairings yielded few candidates for combination of functions into a single instruction; however, it did give interesting insight into the way the system was used.

Looking at the data, a case for combination of functions can only be made for the "get-display" (and "get-view") functions. When obtaining selectors users clearly wanted to check the contents of the selector or to check that the selector had been successfully obtained. Nearly half of the time that users executed "get file" and "get text", they immediately followed them with "display file" and "display text" respectively. The ability to "get and display" would have been very useful. However, this should be an additional function, not a replacement for the plain "get" instruction, because most of the time users followed "get" with an instruction other than display or view.

The combination of "get file and restrict with selector" seems intuitively useful. However, the data showed that only 5% of the "get file" instructions were immediately followed by restrict. This combination could be useful, but is not an important requirement.

Several of the instruction pairings point to a need for better system feedback. The confirmation for a "get" should include the name of the object obtained, so the user can be confident that the transfer has been successful. Keywords for a file or entry should be optionally displayable. A user may not want or need to see the keywords at all times, but should be able to request them.

Looking at the use of the "display entry" and "delete entry" instructions, one sees that the ability to cycle through a file looking at messages in succession, and to go through it acting on messages in an arbitrary order, are both important features. The ability to act on messages in succession using a single function key action is especially important for users who are not accomplished typists.

Finally, it is important to be able to take some actions, such as "delete", "move", "file", "keyword", and "action", on lists of messages as well as on individual messages. This enables a user to handle groups of messages efficiently.

SECTION 5

SELECTION CRITERIA

INTRODUCTION

Sigma provided users with a method of obtaining a subset of a file by using the "restrict" and "augment" instructions. In order to execute these instructions, a user had to specify what criteria were to be used to obtain the subset. Sigma provided 53 different criteria for obtaining a subset of a file. For example, one could get a subset of all the messages originated before a certain date-time-group (DTG). One could get all the messages for which his division or branch had been assigned action or all the SECRET messages in a file. Different types of users needed different criteria for message selection. These criteria could be joined by "and"s and "or"s and modified by "not"s. Restricting a file sometimes returned no messages, one message, or many messages meeting the specified criteria.

There were two ways to select a group of messages from a file on Sigma. One way was to type out the instruction and the criteria to be used. The other way was to use a "saved selector". If a user frequently needed the same set of selection criteria, these criteria could be saved and named. He could then restrict with the named selector to avoid repetitive typing.

This section will discuss three aspects of selection activity on Sigma. First, the different criteria will be discussed and the most frequently used criteria identified. Second, the use of named vs. unnamed (i.e. saved vs. typed) selectors will be presented. Finally, the selectors applied by different types of users will be examined (type of user refers to position in the CINCPAC organization -- action officer, clerk, air desk officer, etc.).

CRITERIA USE

Table 7 shows the frequency of occurrence of different criteria for typed (unnamed) selectors for each user type and for all users. By far the most widely employed criterion was a message's date-time-group. Each user type called for a DTG restriction at least 20% of the time; overall, it appeared in 35% of the selections. The DTG of a message was the most common way of referring to a message in the manual system and the fastest way to retrieve it on Sigma. If one user wanted another to see a message, he would tell him that message's DTG. The second user would then display one of his own files or the datefile (a file of all incoming messages for a particular day) and restrict the file with the desired DTG. Sometimes two or more messages had the same DTG; in that case the user would be shown a list of all these messages and could select the one he wanted. In any case, the requested message was immediately accessible if it had not been archived.

Message subject was the second most frequently used criterion for restricting a message file (17% of the time). Restricting by subject returned all the messages in a given file which contained the specified text string in the subject field. This was handy if a user had to compose a message or briefing on a given subject and needed references or additional information, or if he needed to find messages in his area of responsibility. If a user did not know a desired message's DTG, he would probably know its subject and would use that as the basis for retrieval. Restricting with the subject criterion often returned many messages, since certain words or phrases were widely used in message subjects. However, many messages did not have useful subjects, since the originator did not always specify one. In those cases Sigma attempted to create one, with indifferent success.

"Action <named>" and originator were the next two most frequently used selection criteria (15% and 13% respectively). The action criterion returned all messages for which the named branch had been assigned action, e.g., all the messages designated "action J31". This criterion helped a user locate messages for which he had responsibility, and to use them as references for replies. The originator criterion returned all the messages originated from the specified individual, branch, or base. Both criteria returned a group of messages for the user who needed either the entire group or a single message from that group. This was often useful when a user might not be sure of the DTG, but had some idea of who might have sent the message.

One form of the "restrict" and "augment" instructions permitted users to retrieve all messages except those which matched the specified criteria. This was done using the word "not", e.g., "restrict selection not note" which would return all the messages in the file except those of type note. The "not" capability was used 8% of the time in the unnamed selectors and 8% in the named selectors as well.

Table 7

Unnamed Selectors

User Type

Criterion	AIR	CCCLR	DDO	SUR	DBACT	DBADM	DELLR	DBMGT	EXERC	JRC	J301	NONJ3	J3	JBMGT	ALL
action (resp)	14%			*	*				1%		*	2%			2%
not action (resp)					*										*
action (named)	40%	17%	12%	1%	8%	20%	4%	9%	27%		13%	10%			15%
not action (named)					*						27%				7%
after (dtg)				*	1%				*			6%			*
after (entry)					*										*
autodin					1%				*						*
before (dtg)					*										*
before (entry)									*						*
chopped									*						*
not chopped					*										*
classification	*			1%	*	*	2%								*
cog	8%			*	*										1%
date-time-group	20%	33%	25%	71%	27%	49%	55%	56%	20%	26%	36%	66%	94%	94%	35%
entry no.	7%		13%	5%	7%	6%	3%	12%	6%		2%	10%			6%
not entry no.					*				3%						*
exercise									*						*
not exercise									*						*
from	6%	33%	25%	13%	20%	9%	17%	13%	9%	36%	7%	2%	6%	5%	13%
not from					1%				*	2%	*				*
incoming					*										*
keyword				*	*				6%			2%			*
not keyword					*										*
memo	*				*	*	*	*			*				*
note															
not note	*								2%						*
precedence				*	1%	*	*		7%		*				1%
preparation									*						*
recent					3%										1%
retrieved				*	*		*			5%					*
subject	5%	17%	25%	8%	28%	15%	19%	9%	18%	31%	14%	3%		1%	17%
not subject					1%				1%		*				*
transmitted	*														*
not dtg											*				*

With the exception of DTG, the named selectors used the same criteria as the unnamed selectors as shown in Table 8. 74% of the named selectors used the criteria originator, action or subject or some combination thereof (not orginator/subject, subject/not subject/action, etc.). None of the saved selectors included DTG as a criterion because DTG's were perishable; users did not restrict repeatedly on a particular DTG and so had little need for them in a saved selector. Other criteria used in the saved selectors included precedence, classification, back-copy, unwanted messages (such as weather messages) and keyword.

NAMED VS. UNNAMED SELECTORS

The ability to save selection criteria (as a named selector) was intended to save users from typing when the criteria consisted of many items and/or were frequently used. Many users found this capability useful; 32% of the selections were done with saved selectors. Many of these selectors contained multiple criteria; a few specified up to 50 or 60. For these users, the saved selector capability was particularly beneficial since it saved the inevitable time and mistakes associated with entering and executing lengthy instructions. However, saved selectors could not be edited, a capability that would have been useful. When users wanted to change some criteria in a saved selector, they had to recreate it in its entirety.

J301, the branch responsible for message distribution within J3, accounted for 50% of all named selector use. They made up saved selectors for each branch specifying the types of messages to be sent to that branch, using originator, subject, action, and other criteria. By doing this, the saved selector would return a group of messages with common routing characteristics and the J301 message router could then distribute that subset with a single "route" instruction. The saved selector capability considerably simplified the process of message distribution.

USER TYPES AND CRITERIA USE

As shown in Tables 7 and 8, the data on criteria usage are presented by user type (position in the CINCPAC organization) to identify different patterns (if any) of selection criteria use. The heaviest users of selectors were the Command Center users (Air and Surface officers in particular), exercise users, action officers, and J301.

Table 8

Named (Saved) Selectors

User Type

	AIR	CCCLR	DDD	SUR	DBACT	DBADM	DBCCR	DBMGT	EXERC	JRC	J301	NONJ3	J3	J3MGT	ALL
action (named)					*										*
action/from					*										*
action/subject							41%				9%				4%
action/from/subject					26%	100%	54%			*	41%				29%
a bad copy/action					4%										1%
from					3%					12%					2%
from/subject					20%		5%				35%				24%
garbage or trash or unwanted	98%				10%					85%					20%
keyword															
precedence	*			100%	*					*					*
precedence/classification					14%										5%
subject	2%				22%						15%				15%

* - .05%

- including "not"s

The Command Center users, primarily air and surface desk officers, used a variety of criteria in restricting message files with typed selectors. Both air and surface desk officers used the date-time-group, subject, from, and entry number criteria frequently. The air desk officers used the action <named> criterion more than any other. This may have been because the air desk was responsible for preparing the daily readboards. Messages for action to J3 would go in one section of the readboard while other messages were put in other sections. In addition, the air desk was responsible for reviewing and sometimes distributing high precedence incoming message traffic. The action criterion was very useful for this task as well.

Although some air desk officers used named selectors to weed out unwanted messages (e.g., weather messages), neither air nor surface desk officers used them extensively. (The selection criteria used for readboard creation were simple and convenient to retype everyday.) The air desk received copies of all incoming messages for J3 but was only interested in those relating to Command Center operations or high precedence messages which arrived after normal business hours. Upon logging on, these officers restricted the file using a "trash" selector (describing unwanted messages) and deleted all these messages.

The exercise users employed many different criteria in selecting message subsets during the March 1979 exercise. The most popular criterion was "action <named>" (27%). Due to the large volume of messages generated during the exercise, this criterion was useful for message distribution and for retrieving messages for which exercise officers were responsible.

As for most types of users, during the exercise the other most frequently used criteria were date-time-group, originator, and subject. Exercise officers did not use saved selectors at all since they used the system for only a few weeks. During a real crisis situation, saved selectors could significantly speed up message retrieval; users would be able to access quickly messages of importance.

Action officers used "restrict" and "augment" instructions extensively with both named and unnamed selectors. The most frequently used criteria were, as for most users, date-time-group (27%), subject (28%), from (20%), action <named> (8%) and entry number (7%). They also restricted files based on other criteria such as before/after dtg, precedence, classification, and recent.

Action officers found the saved selector capability useful; each division or branch created its own selectors enabling them to access quickly the messages it was responsible for. Generally, these messages were the ones it was assigned action on, those concerning a particular subject, or those from a particular command or individual. 71% of the saved selectors used by action officers consisted of some combination of these three criteria (action, subject, from).

J301 was the branch responsible for message distribution in J3. Most incoming messages arrived at J301 first. They then determined who should be assigned action on messages and who should receive them for info. To determine how a message should be routed and who should have responsibility for it, the criteria from, subject and action were most commonly used. Together, these three criteria were used (in some form and combination) in 61% of the typed selectors and in all the named selectors. As pointed out in the previous subsection, the J301 users used saved selectors to obtain a group of messages with common routing characteristics, and then routed them as a group.

CONCLUSIONS

It is clear from these data that CINCPAC users found the ability to subset message files by specifying certain criteria a very useful feature of Sigma. In particular, the criteria date-time-group, subject, originator, and action designee were widely used and should be included in any future message handling system. CINPAC users also benefitted from the ability to name and store previously defined selectors, saving them much typing and many errors in subsequent use. This, too, would be a welcome feature in future systems.

SECTION 6

SELECTION LEVEL

DISCUSSION

The "restrict" and "augment" instructions described in the previous section could be used sequentially to obtain a subset of a subset or to combine two subsets of a message file. First, a user would obtain a subset using a "restrict" instruction. If a second "restrict" was done, the subset was restricted further resulting in a second subset. If an "augment" was done after the initial "restrict", it searched the original file using the criteria in the "augment" instruction and added those messages to the first subset. This process could continue several times until the user obtained the desired set of messages. Each "restrict" or "augment" instruction could use either a typed selector or one previously defined, named, and saved. Each selector could be a single logical criterion or a compound set of criteria linked with "and", "or", "not", and parentheses.

This multiple "restrict" and "augment" activity is the subject of this section. Of interest is the level of restriction and augmentation normally required before the user obtained the desired set of messages -- that is, the number of times he issued consecutive "restricts" or "augments" before resetting or closing a message file.

Table 9

Selection Level Occurrence

Depth of Selection	Occurrence
1	12,644
2	623
3	50
4	14
5	2
6	1

The occurrence of the various levels is shown in Table 9. (Levels include all combinations of consecutive "restrict" and "augment" instructions.) Note that of the 13,334 selection sessions, 12,644 (95%) used only one level of selection. Two levels were required about 5% of the time with the remaining 0.5% requiring 3-6 levels of selection. This indicates that users were able to predict accurately what criteria were necessary to obtain the message(s) they needed.

The level 1 data may be somewhat misleading because they include the saved selectors which made up 39% of the selections. Many saved selectors incorporated several selection criteria, and the data does not reflect this complexity.

One level of restriction was usually sufficient to obtain the desired set although occasionally users found the capability of further restriction useful. It was certainly useful to be able to save selectors; instead of issuing several "restrict"s and/or "augment"s, once the saved selector was prepared, only one instruction was necessary to obtain the effect of several levels of restricting. In compound selectors, the logical operators "and", "or", "not", and parentheses did the work of additional "restrict"s. Although data on compounding of typed selectors were not extracted from the session transcript data because of the difficulty of translating and analyzing logical constructions, a spot check of 200 "restricts" showed that 191 of them used only a single criterion.

SECTION 7

THE "RESTORE" INSTRUCTION

INTRODUCTION

On Sigma, when a user no longer needed a copy of a message, file, text object or selector, he could use the "delete" instruction to remove it from his directory or file. If, after issuing a "delete" instruction and before logging off (or closing a message file), a user found he really needed the deleted object, he could issue the "restore" instruction. This restored the deleted object in the user's directory. The utility of the "restore" instruction for automated message handling will be examined in this section.

DISCUSSION

Table 10 shows the number of objects deleted and restored throughout the experiment (22 February 1979 - 29 September 1979). Note that 4.6% of deleted files were restored, 1.8% of deleted text objects were restored and 0.2% of deleted messages were restored. No selectors were restored.

Table 10

Items Created, Deleted, and Restored

	Number Created and Acquired	Number Deleted	Number Restored	
Messages	N/A	373042	3048*	0.8%
Files	896	669	31	4.6%
Text Objects	886	600	11	1.8%
Selectors	333	159	0	0.0%

* The number of messages restored per instruction was estimated from DCF data.

The low percentage of messages restored was due to the fact that, because of Sigma's central message storage mechanism, when a user issued a "delete message" instruction, he was deleting only the pointer to that message, not the message itself. Pointers to each message were available in the datefiles, a daily log of the incoming message traffic. Thus if a deleted message was needed, the user could either "restore" it or copy it from the datefile into his own files. Users could delete groups of messages, knowing that they could always copy any that should not have been deleted.

The high number of messages deleted compared to the number created can be accounted for by the incoming message load; during FEU approximately 139,000 incoming messages were delivered to Sigma, and pointers to each message could be distributed to more than one user.

No selectors were restored during the experiment. Most selectors could be recreated using the "restrict" and/or "augment" instructions. It would have been painful to recreate saved selectors with many levels of selection criteria; presumably this made users careful not to delete them.

Table 11 shows the number of times a "restore <object>" instruction was executed immediately after a "delete <object>" instruction, restoring the newly deleted object. More than 60% of

SECTION 10

TRACKING USERS

INTRODUCTION

One potential area of interest in the experiment was the possibility that a user's approach to an automated message handling system might grow in sophistication as he became more experienced in using the system. In order to explore this possibility, six representative subjects were chosen. These six each represented a different sector of the experiment population, and each was a fairly active user throughout FEU.

Once the users were selected, data were reduced for each for all fifteen two-week periods which made up FEU. These data include number and type of instructions executed, errors (or flags) which occurred, and amount of time spent on-line. In addition to these data, Figure 3 presents data on system up-time for comparison with user on-line time; one of the reasons for low user on-line time during some periods could have been low system availability.

USER A

User A was a watch officer in the Joint Reconnaissance Center (JRC), which was responsible for monitoring reconnaisance missions. Figure 4 shows his use of the system for each of the two week periods of the experiment. This figure shows the amount of time he spent on-line during each period, the total number of instructions he executed, and the number of errors which occurred per 100 instructions executed. Table 15 shows the specific Sigma instructions executed during each period. As Figure 4 demonstrates, A's Sigma activity varied greatly during FEU. During the first two weeks, he was on-line about 54 hours and executed over 900 instructions. Toward the end of the experiment, A was executing only 100-200 instructions per period. By that time, overall use of the system had dropped off somewhat; some of the users began to rely again on manual procedures in anticipation of Sigma's removal. Another reason that activity dropped near the end was that some users left CINCPAC and their successors were reluctant to learn a system that was only going to be around for a short time. Also, many users took leave during the summer.

managers of J3, the importance of readboard access is proportionately greater than the relatively small number of minutes spent looking at them. Future message handling systems would do well to make it easy for all users to have ready access to special files of this sort.

because he never displayed the on-line version, since he rarely used Sigma. However, his clerks displayed it in his behalf and he occasionally looked at it on their terminals; that usage was ıecorded as "admin/clerıcal".

Action officers were moderate users of the readboards (2-3 minutes per day). It is interesting to note that action officers with lower incoming message loads (both action and information messages) displayed the readboards much more often than action officers with heavier incoming message loads. Action officers with lighter message loads were often planners, not as involved with day-to-day operations, and probably had more time to browse through readboards than users with high incoming message loads.

Administrative personnel were heavy readboard users. It is likely that much of their use was on behalf of the division and branch chiefs. These users would search the readboards for messages of particular interest to their superiors and print copies of them. Some of the division and branch management data represent clerks logging on in their chief's accounts and printing messages of interest. Many division chiefs preferred handling paper copies of messages and seldom used Sigma.

The Command Center users (DDO and air desk officers) were not heavy readboard users. The DDO displayed readboards early in FEU but did not continue. The air desk officers, creators of the readboards, displayed them frequently at first, but their use tapered off. Perhaps at first they displayed the readboards to make sure they had been created properly. Once they gained confidence in Sigma and their readboard creation procedures, they no longer needed to display the readboards as often. As with many first-time automated system users, they may have distrusted Sigma initially; displaying the readboards was a way of checking up on the system.

CONCLUSIONS

Use of the readboards (2-5 minutes per day) by the action officers, administrative personnel, and division/branch chiefs indicates that the readboards were of interest to users other than the director and his immediate staff. The readboards kept them informed of CINCPAC's activities, particularly those of special interest to the Director (J3). Although the number of minutes per day spent looking at the readboards does not in itself appear very high, the division/branch chiefs found the information they got from them very valuable. Since these chiefs were the middle level

Table 14

Readboard Use

Period I: 18-26 May, 1979
Period II: 9-21 July, 1979
Period III: 16-29 Sept, 1979

AIR

	I	II	III
total minutes	416.5	26.8	63.4
user-days	74	13	41
mean min. used per user-day	5.6	2.1	1.5

DDO

	I	II	III
total minutes	239.1	-	-
user-days	21	-	-
mean min. used per user-day	11	-	-

ACTION OFFICERS

	I	II	III
total minutes	308.3	328.9	132.8
user days	110	145	65
mean min. used per user-day	2.8	2.3	2.0

ADMIN/CLERICAL

	I	II	III
total minutes	174.2	205.8	306.1
user-days	38	39	55
mean min. used per user-day	4.6	5.3	5.6

DIVISION/BRANCH MANAGEMENT

	I	II	III
total minutes	133.9	61.2	22.0
user-days	53	45	25
mean min. used per user-day	2.5	1.4	0.8

TOTAL

	I	II	III
total minutes	1272.0	622.7	524.3
user-days	296	242	186
mean min. used per user-day	4.3	2.6	2.8

SECTION 9

READBOARD USAGE

INTRODUCTION

The readboards on Sigma were files of messages of special interest to the Director of Operations (J3). They were prepared by air desk officers in the Command Center on the night shift so that the Director could be quickly apprised of recent developments upon his arrival in the morning. Prior to Sigma's installation, readboards consisted mostly of hard copies of messages, plus a few other items on paper. There was only one copy, and it was available only to the Director and his immediate staff. With Sigma, the readboards were available to all users, although the initial effort of getting access to them was quite complex and often required help from the experiment staff. This section describes the use of the on-line readboards during FEU.

Although readboard capabilities were available throughout FEU, the data on readboard usage were studied for three periods: 18-26 May, 1979; 9-21 July, 1979; and 16-29 September, 1979, as reported in Table 14. These data cover the use of all readboards (they were divided into five files) and are grouped by user type.

USE OF READBOARDS

Five groups of users displayed readboards: air desk officers; the Duty Director of Operations (DDO) in the Command Center; action officers; administrative personnel (including clerks); and division/branch chiefs. For each group, the total minutes spent displaying readboards and the number of user-days during which readboards were displayed are shown for each period. The mean minutes readboards were used per user-day is also shown.

The primary users of the on-line readboards were action officers, clerks and administrative personnel, and division/branch chiefs, all of whom had not previously had access to them. They used readboards throughout FEU. In particular, readboards gave them a chance to see what the Director was seeing, and thus be better prepared to respond to his requests. The Director (J3) continued to use the paper version of the readboard, which was also prepared by air desk officers. The Director's data are not included in Table 14

Even after a message was archived, some information about it had to be kept on-line. This information was needed by the archive retrieval program to determine which tape to mount and where on the tape to look for the message. Since Sigma was an experimental system with a limited life, this extra space was no particular burden. In a permanent system, the amount of space required to store archive information might eventually become a problem. Sigma experience suggests that a second threshold of about 200 days could be established, after which all information about older archived messages could be removed from on-line storage. At most installations, users would still be able to retrieve older messages from their local telecommunications center.

Two other possible features of an archive facility are "Archive Early" and "Keep in On-Line Storage". Sigma provided early archive; the operations staff could designate certain types of messages which were archived sooner than others. At CINCPAC, weather messages and Foreign Broadcast Information Service (FBIS) messages were archived after one day. This provided space needed to keep other types of messages on-line longer. The ability to exclude messages explicitly from being removed from on-line storage was not made available to users. Many of them suggested that it would have been a useful feature, but Sigma's designers were afraid that the privilege might have been abused, leading to severe storage problems. Some users took advantage of the fact that if they looked at a message every couple of weeks, it was not removed from on-line storage.

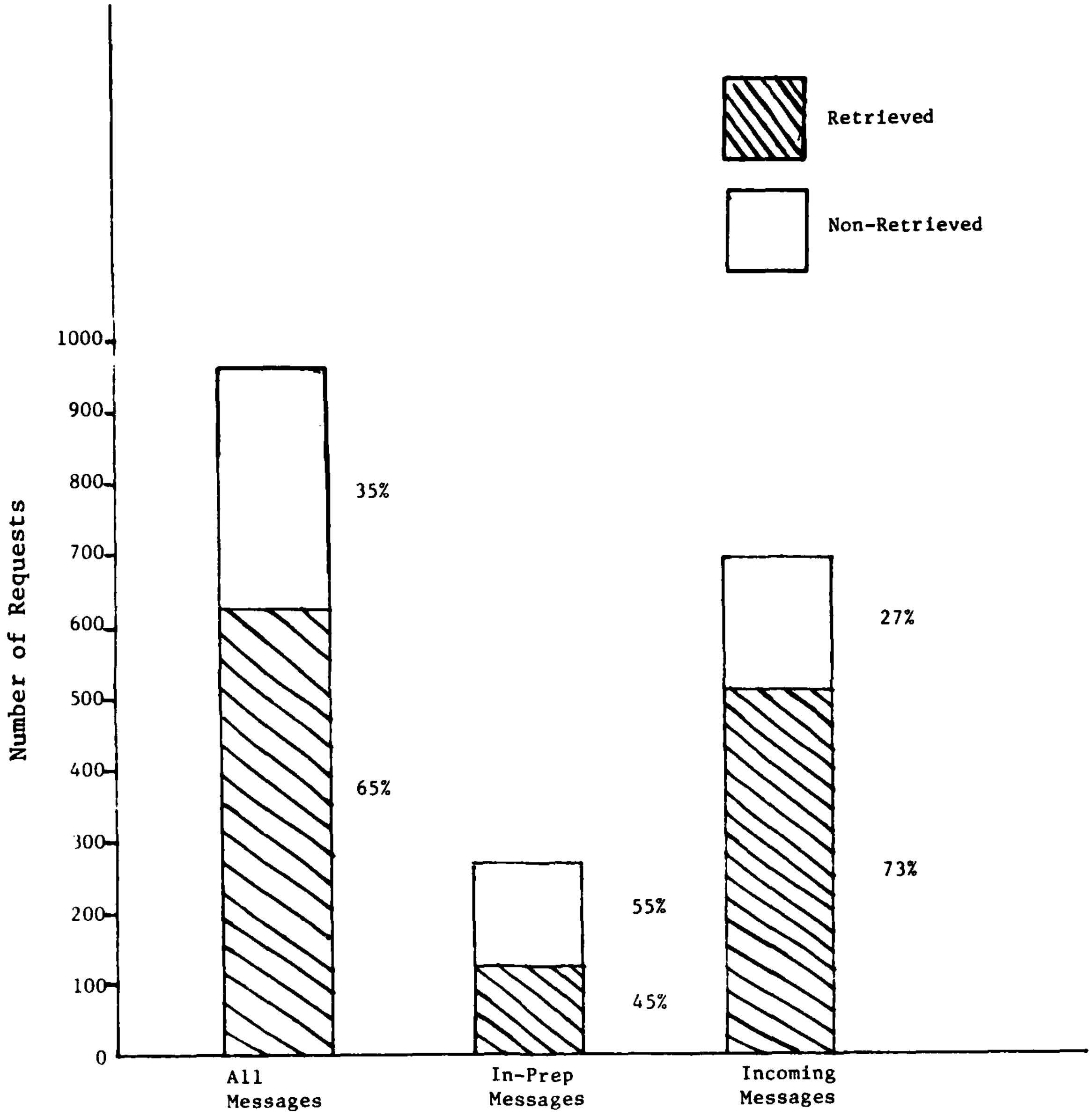


Figure 2: Retrieval of In-Prep and Incoming Messages

AUTODIN VS. IN-PREP MESSAGES

Over 70% of the messages requested were incoming AUTODIN messages (including backcopies of outgoing messages). The remainder of the requested messages were in-prep messages, those that had been prepared using Sigma. Some of them had been released; others not. There were many more AUTODIN messages than in-prep messages in Sigma's message space; about 95% of the messages were AUTODIN. The average age of requested AUTODIN messages was about 48 days; that of requested in-prep messages about 65 days.

Figure 2 shows the breakdown of all requested messages by message type (AUTODIN vs. in-prep). In addition, this breakdown is shown for AUTODIN messages only and for in-prep messages only.

In-prep messages made up about 28% of the total messages requested. Of that 28%, only about 45% were retrieved. In contrast, 73% of the requested AUTODIN messages were retrieved. One reason for this difference might lie in the reasons for which users retrieved the two message types. Often the reason for retrieving in-prep messages was to use them as a guide or template in composing a potential outgoing message on Sigma. When a user found out that he might have to wait to get the older message, he may have often decided to go ahead without it, since its retrieval was not essential to the process of creating a new message. In the case of AUTODIN messages, however, there often was no reasonable alternative to retrieval.

CONCLUSIONS

No system has unlimited on-line storage space, so eventually older messages must be removed to make way for newer ones. On Sigma, the age threshold for archiving ranged from ten to thirty days. Increasing the threshold to thirty days reduced the number of retrievals from six per day to three or four per day. When the threshold was ten days, the majority of messages requested were from eleven to fifty days old. After the threshold was increased to thirty days, the majority of messages were from fifty to 150 days old.

In selecting a range of thresholds for a message handling system, the needs of the user community must be considered. The oldest message requested on Sigma during FEU was 295 days old. Over 95% of the messages were less than 125 days old.

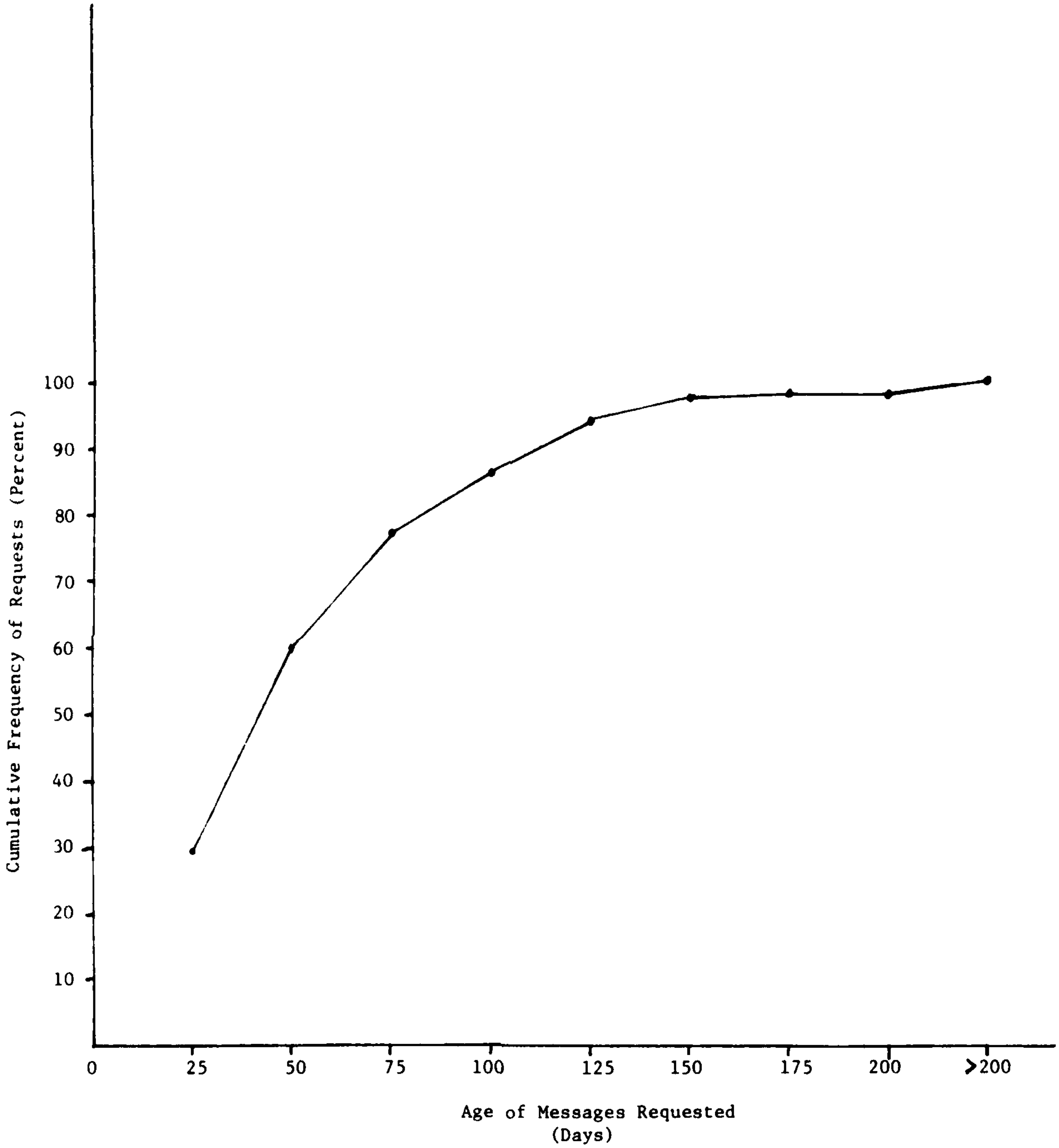


Figure 1: Age of Requested Messages

Tatle 13

Message Retrieval: In-Prep and Incoming Messages

In-Prep Messages

Retrieved Messages

Threshold (Days)	# Days at Thresh.	# Requests	Mean #Requests Per Day	Mean Message Age (Days)
10	76	44	0.6	47.8
11 - 29	25	14	0.6	36.0
30	119	63	0.5	89.0

Non-retrieved Messages

10	76	79	1.0	48.3
11 - 29	25	25	1.0	69.6
30	119	42	0.4	85.5

Incoming Messages

Retrieved Messages

10	76	242	3.2	19.8
11 - 29	25	33	1.3	25.5
30	119	227	1.9	72.8

Non-retrieved Messages

10	76	71	0.9	21.8
11 - 29	25	14	0.6	18.7
30	119	102	0.8	65.8

Table 12

Message Retrieval: All Messages

All Messages

Threshold (Days)	#Days at Thresh.	#Requests	Mean #Requests Per Day	Mean Message Age (Days)
10	76	436	5.7	33.7
11 - 29	25	86	3.4	38.9
30	119	434	3.6	74.7

Retrieved Messages

10	76	286	3.8	32.6
11 - 29	25	47	1.9	28.6
30	119	290	2.4	76.3

Non-retrieved Messages

10	76	150	2.0	35.7
11 - 29	25	39	1.6	51.4
30	119	144	1.2	71.5

The threshold was gradually lengthened during the experiment after an increase in disk storage. For the first 76 days of FEU, the threshold was ten days. Over the next 25 days, it was gradually increased from 11 to 29 days. For the last 119 days of FEU, it was maintained at thirty days. The second column of Tables 12 and 13 show how many days the threshold was kept at each level.

The third column of Tables 12 and 13 gives, for each threshold, the number of display requests that were made for archived messages. The fourth column shows the mean number of requests per day. The fifth column, labeled "Mean Message Age", is the total age of all messages divided by the number of requests.

The Data

More than half the requested messages were actually retrieved, regardless of threshold. Overall, users requested that 65% of the messages be retrieved. In the case of non-retrievals, it is likely that users decided that seeing the archived message was not worth the delay necessary for its retrieval. It normally took 15 minutes or so to retrieve a message from the archive. However, if a message was needed urgently the user could call Sigma operations and request an immediate retrieval. In addition, watch standers in the CINCPAC Command Center (who were responsible for monitoring incoming traffic during non-duty hours for the rest of the CINCPAC staff) often browsed through their files pressing the "Display Next Entry" key. They were not always interested enough in the message to wait the necessary time, and perhaps felt that seeing the message was not worth the effort that would have to be expended by Sigma operations personnel in retrieving it.

Figure 1 presents a cumulative frequency of the age of requested messages. Note that of the 956 messages requested, 13% were over 100 days old; 2% were over 200 days old.

As might be expected, as the threshold increased so did the average age of requested messages. Since messages were kept on-line longer due to the higher threshold, the messages were, on the average, older. This was true for both retrieved and non-retrieved messages.

SECTION 8

RETRIEVAL OF MESSAGES FROM THE ARCHIVE

INTRODUCTION

In order to conserve Sigma's on-line storage space, messages were moved ("archived") onto magnetic tape when they had not been used for a certain length of time. Pointers to the archived messages were kept in Sigma's message storage space. When a user saw a description of a message (called an "entry") in one of his message files, he did not know whether it had been archived. If a message he attempted to display had been archived, he was presented with the response "That message has been archived. Do you want it retrieved?" The user could respond by pressing either the "Yes" or "No" keys. This section will discuss various factors, such as type of message and age of message, which may have affected the user's decision about whether a particular message should be retrieved.

RETRIEVED VS. NON-RETRIEVED MESSAGES

The Data Tables

Tables 12 and 13 present data on all archived messages which users tried to access. In Table 12, the section labeled "All Messages" refers to all requests for archived messages. The section labeled "Retrieved Messages" shows the number of times users responded "Yes" to Sigma's retrieval inquiry, and the section labeled "Non-Retrieved Messages" shows the number of times users responded "No". Table 13 divides the same data into two groups: AUTODIN messages, including both incoming messages and comeback copies of outgoing messages; and in-prep messages (drafts of messages prepared by Sigma users, which may or may not have been released).

The first column of both Tables 12 and 13 is labeled "Threshold". This was the number of days that elapsed from the time that a message was last examined by a user until the time that it was archived. For example, if a message was last looked at eight days after its arrival and the threshold was ten days, it would be archived when it was eighteen days old. This threshold was controlled by operations personnel and was a function of available on-line storage space.

the "restore entry" instructions were issued immediately after "delete entry" instructions. Since several messages could be deleted (and restored) at a time, it is likely that a user deleted a group of messages and then discovered that he wanted to keep one or more of those messages.

Note that only a few "delete" instructions were followed immediately by a "restore" instruction. This was true for messages, text objects and selectors.

Table 11

Number of Immediate Restores

Instruction Pair	Occurrences	Restore Instructions
delete entry/restore entry	489	762
delete file/restore file	6	31
delete text/restore text	4	11

CONCLUSIONS

The utility of a "restore" or "undelete" instruction is clear for any computer system which provides a "delete" capability. (A "delete" capability is necessary to prevent storage space from becoming overloaded.) It helps undo the inevitable mistakes that users, particularly novice users, make. Sigma users found its "restore" instruction useful, particularly for restoring files and text objects which would have required considerable time and effort to recreate.

If a "restore" function were not provided on a system, it would be necessary for the system administrators to attempt to retrieve a deleted document from the system storage. Such a process, if possible, would be inconvenient for both the users and administrators. To prevent this, any automated message system should provide an "undelete" capability. System backup and retrieval procedures are still needed, but the ability to restore recently deleted objects via an instruction will relieve the load on the operations staff.

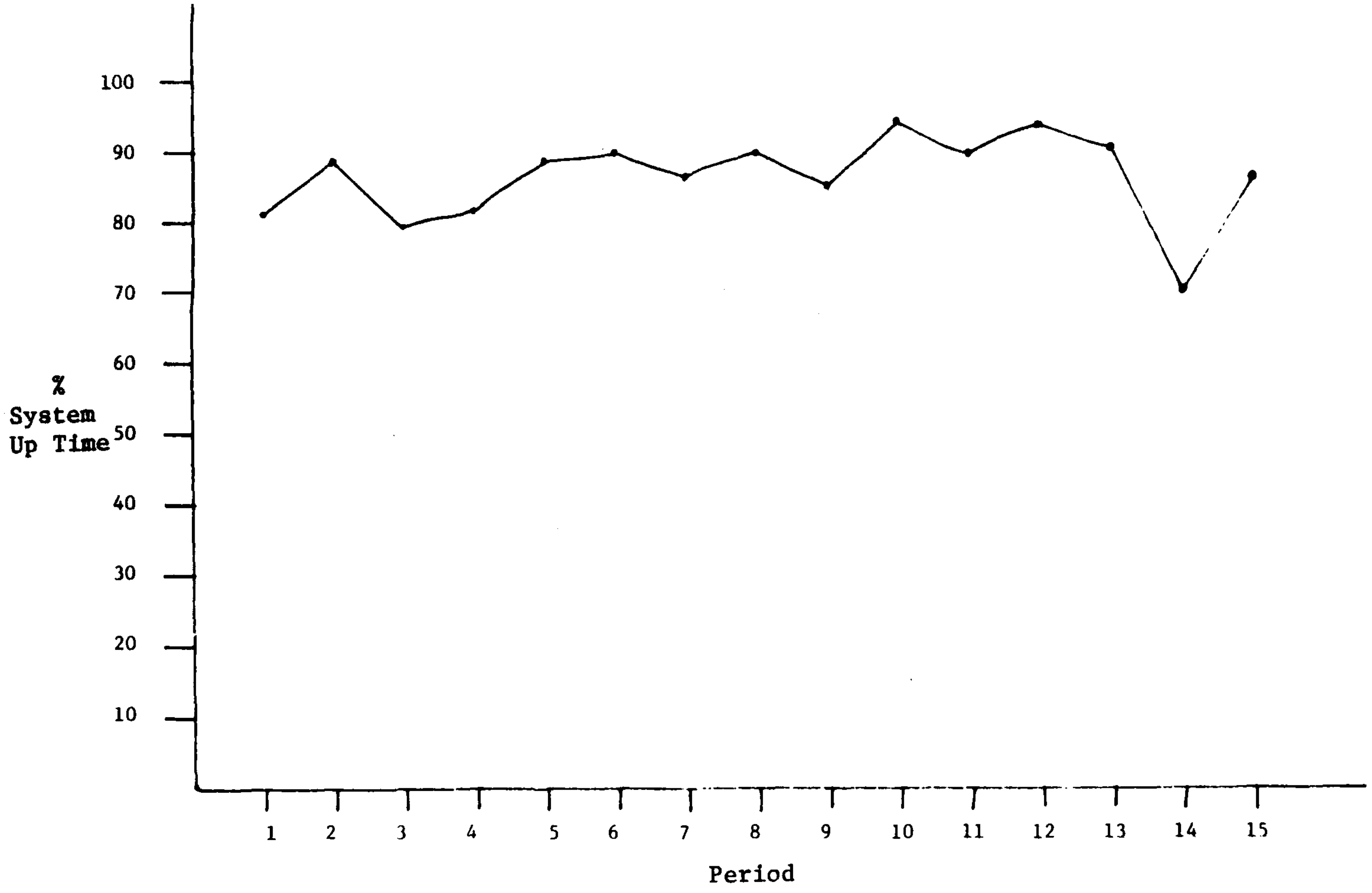


Figure 3: System Up Time

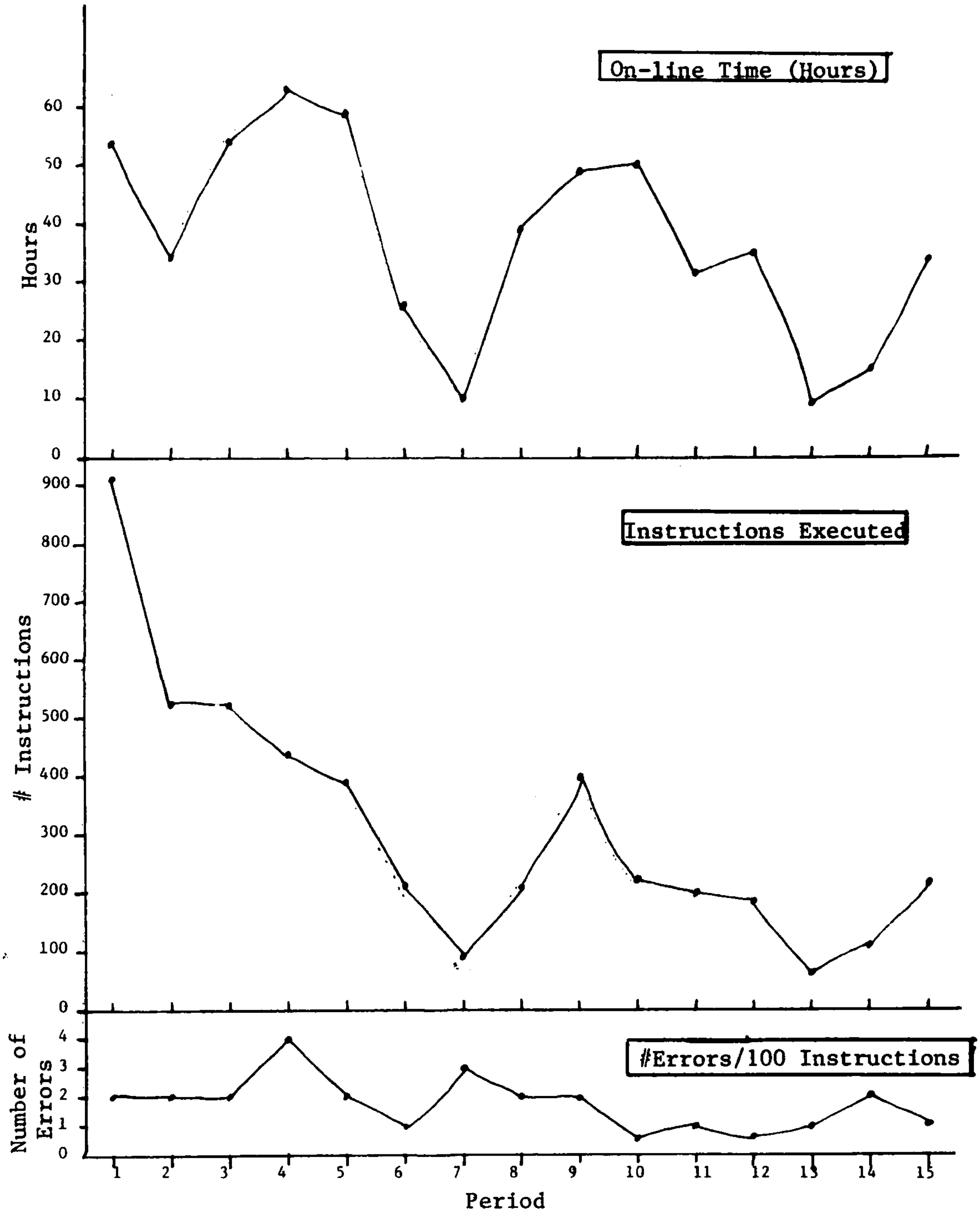


Figure 4. User A: Sigma Use

Table 15

User A: Instruction Use

Instruction	Period 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
display message		x	x	x	x		x		x	x	x	x	x		
display file	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
print	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view directory	x	x	x	x	x			x	x		x				
clear view window	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
save/update	x	x	x	x	x	x		x	x	x	x	x		x	x
finish	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
file/move	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
delete message	x	x	x	x	x	x		x	x	x	x	x	x	x	x
display text	x	x	x	x				x	x	x	x	x		x	
go to	x	x													
create text	x	x	x	x		x		x	x		x				x
alert	x			x	x			x	x	x	x	x			
find entry			x		x										
restrict/augment			x	x	x	x	x	x	x	x	x	x	x	x	x
backup			x	x	x	x	x	x	x	x	x	x	x	x	x
forward message			x	x	x		x	x	x		x	x			x
comment			x	x		x									
chop message			x												
create message			x	x	x		x				x		x		
create file			x			x					x				
coordinate					x		x						x		
copy/move...text							x				x				
find string													x		

A's activity was high (particularly his instruction use) during the first two weeks of FEU. In periods 7 and 13 both his on-line time and instruction use were very low. This could be attributed to two factors. First, he may have been on leave for part of the time, or, second, he could have been working temporarily in J314, the branch responsible for reconnaissance mission planning, as most JRC officers occasionally did. This branch had only one terminal available, and it was customary for J314 users to log on, do their work, and log off, rather than to stay logged on for extended periods as JRC users did.

Figure 4 also gives data on A's errors. His error rate was fairly constant throughout the experiment, hovering at about 2 per 100 instructions. While one might expect the error rate to decrease as the user became more familiar with the system, A was experienced when FEU began; his use had presumably already stabilized when data collection began.

The instructions A executed during FEU are shown in Tabl 15. The core of his activity included the instructions "display message", "display file", "view message", "print", "finish", "file/move", and "delete message". A few weeks into the experiment he began using some of Sigma's other message handling features such as "restrict/augment", "create message", and "coordinate message". His message creation activity probably occurred while he was working in J314; message creation was infrequent for watch standers.

USER B

User B was a branch chief. Figure 5 and Table 16 present data on his Sigma activity during FEU. As Figure 5 shows, he did not begin using the system until eight weeks into FEU and was on leave during period 6.

At first, B had a low activity level. As he became more familiar with the system his use increased. During period 10 it peaked at 30 hours on-line and and at about 800 instructions for the two weeks. During this period system availability was its highest.

B's error rate decreased early in his use of Sigma; his activity level increased dramatically during this period. As his proficiency improved, his error rate dropped. The number of errors he caused averaged less than 1 per 100 instructions later in the experiment.

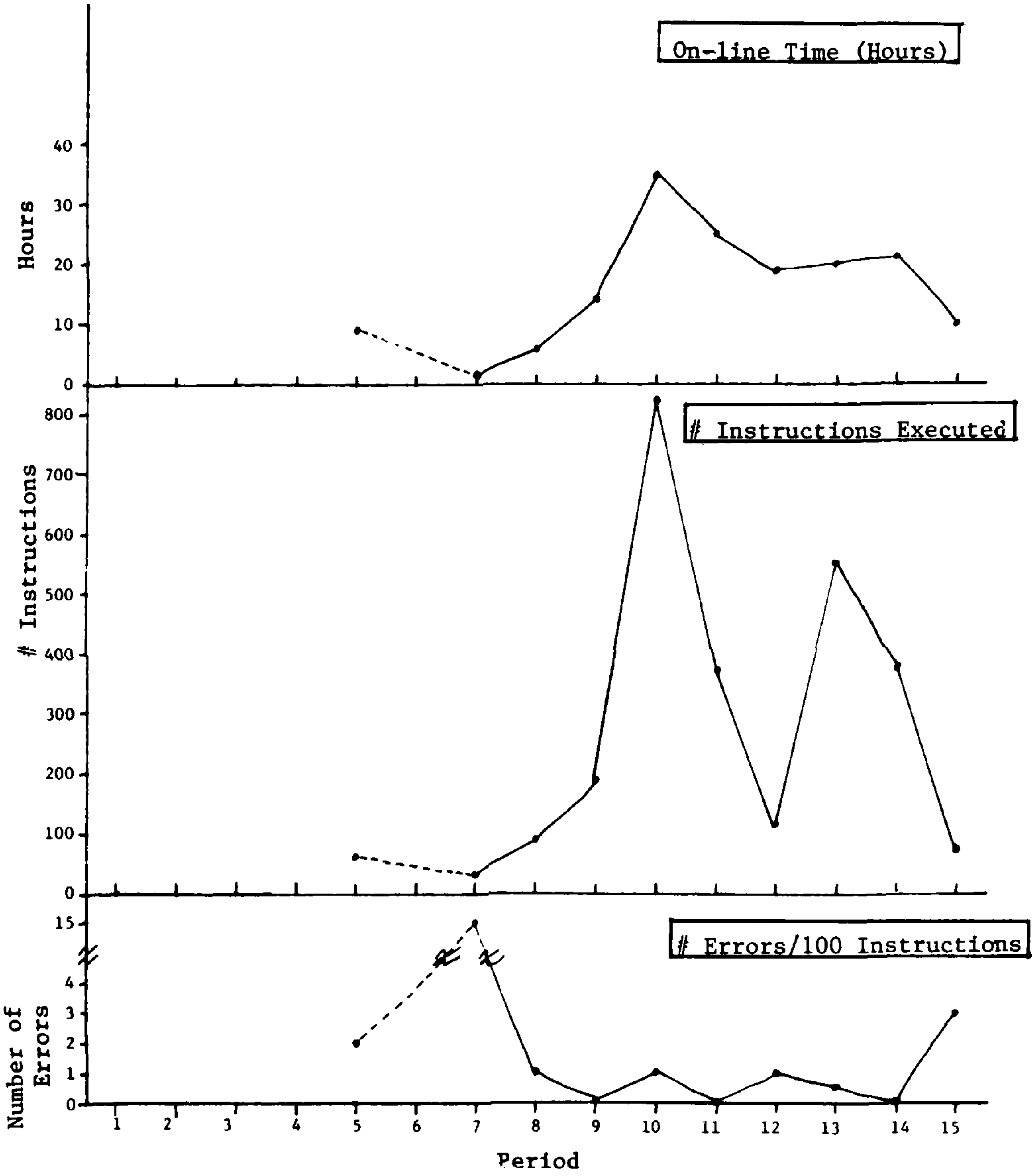


Figure 5. User B: Sigma Use

Table 16

User B: Instruction Use

	Period														
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
display message	-	-	-	-	x	-	x	x	x	x	x	x	x	x	x
display file	-	-	-	-	x	-	x	x	x	x	x	x	x	x	x
print	-	-	-	-	x	-	x	x	x	x	x	x	x	x	x
view directory	-	-	-	-	x	-				x					
clear view window	-	-	-	-	x	-	x	x	x	x	x	x	x	x	x
file/move	-	-	-	-	x	-				x	x				x
delete message	-	-	-	-	x	-				x					
restrict/augment	-	-	-	-	x	-	x	x	x	x	x	x	x	x	x
backup	-	-	-	-	x	-						x			
find entry	-	-	-	-		-		x	x	x	x	x	x	x	
finish	-	-	-	-		-				x	x	x		x	x
highlight	-	-	-	-		-				x					
view message	-	-	-	-		-						x			

Table 16 demonstrates that B executed a small but fairly consistent set of instructions throughout the experiment. He executed only those instructions necessary to review his incoming messages; his responsibilities as branch chief limited his opportunity to take full advantage of Sigma's capabilities. If he needed information from the system, he asked one of the clerks to get it for him. His most frequent instructions were "display message", "display file", "print", "clear view window", "restrict/augment", and "find entry". Other instructions, including "view directory", "file/move", and "delete message", were used only occasionally.

USER C

User C was an air desk officer in CINCPAC's Command Center. He was responsible for monitoring incoming messages, giving special attention to those dealing with air operations. Air desk officers were also charged with preparing message readboards for the Director of Operations (see Section 9). C began using Sigma almost a year before FEU began; his use during FEU is shown in Figure 6 and Table 17.

Figure 6 shows that C's activity level was high and fairly stable throughout most of FEU (periods 3-11, 13-15). As with User A, the graphs for the two periods are similar, indicating that the number of instructions per hour was somewhat constant. This activity pattern was typical of Command Center officers who often left their terminals logged on throughout their watches. The primary concern of watch standers in the Command Center was emergency situations; until these situations arose the watch standers sometimes were not very busy. During these relatively unhurried periods they browsed through message files and created text objects. This air officer's level of use average 40-70 hours and 600-1000 instructions per two-week period.

C's activity dropped dramatically during period 12 (to 18 hours and 270 instructions). He may have been on leave for part of this period. Terminal availability in the Command Center was not a problem; the air desk had its own terminal.

During the first two-week period C's activity level was much higher than at any other time during FEU. Investigation showed that this was because the air desk officers who followed him on duty did not log in as themselves when they began their shifts; they used the air desk's Sigma account under C's login name.

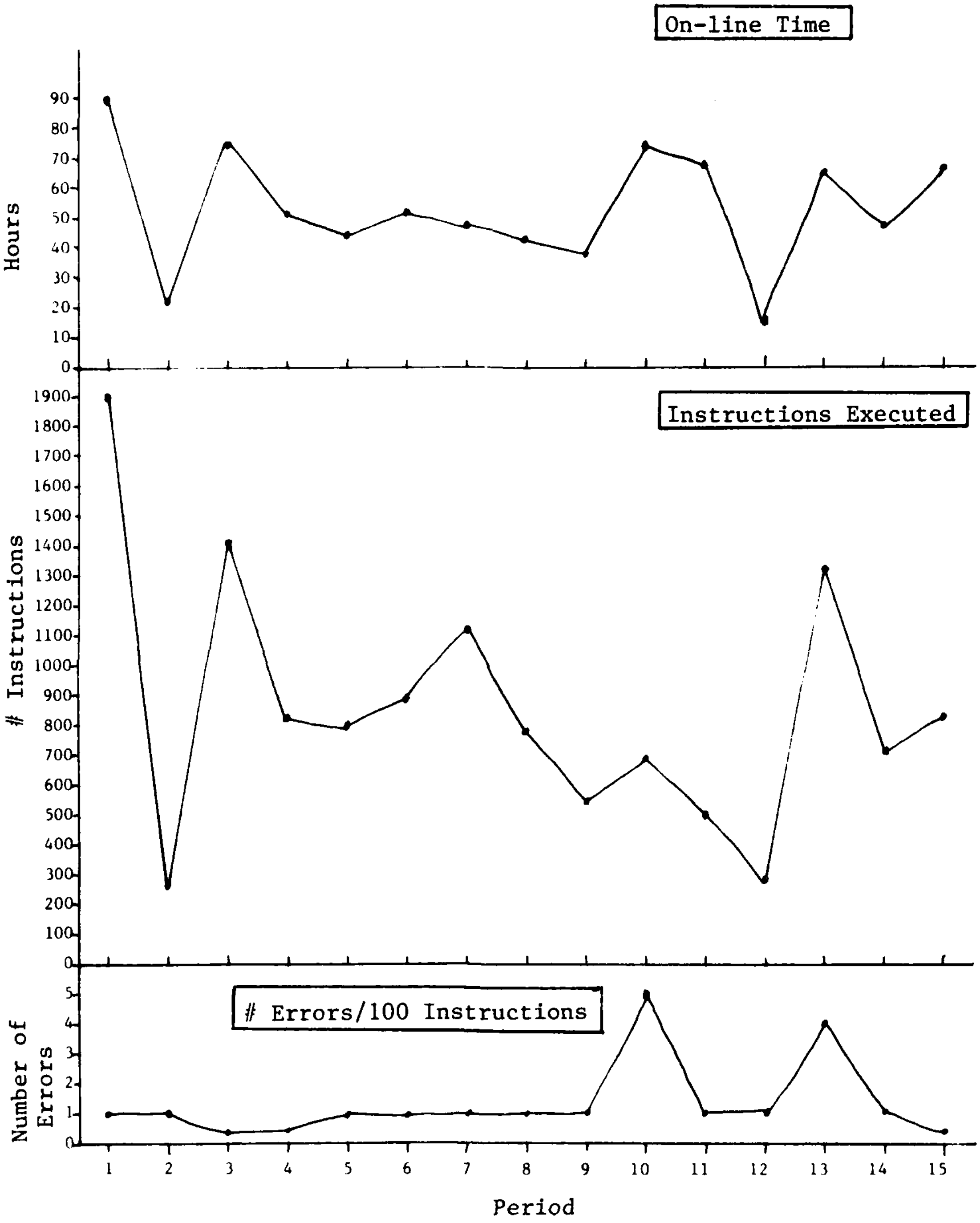


Figure 6. User C: Sigma Use

Table 17

User C: Instruction Use

Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Period														
display message	x		x	x	x	x	x	x	x	x	x	x	x	x	x
display file	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
print	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view directory	x									x					
clear view window	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
save/update	x					x	x	x	x	x	x		x		x
finish	x	x		x	x		x			x	x				
file/move	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
delete message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
restrict/augment	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
backup	x	x	x	x	x	x	x	x	x	x	x		x	x	x
display text	x		x	x	x	x	x	x	x	x	x		x	x	x
alert	x	x							x		x		x	x	x
find entry		x													
copy/move...text							x	x			x				x
empty/sort file								x	x	x	x	x	x	x	x
create file										x					
create selector											x				
view selector											x				
go to													x	x	
readdress													x		

C's error rate also remained fairly constant during FEU. For most of the two week periods he made only one error per 100 instructions. The exceptions were periods 10 and 13 when he caused [illegible] and 4 errors, respectively, per 100 instructions. In period 13, he added two new new instructions to his repertoire. His unfamiliarity with these instructions may have caused him to make more errors than usual.

USER D

User D was the chief of a very small branch. He was introduced to Sigma more than a year before FEU began. As Figure 7 shows, his activity level remained stable and high throughout the experiment. He averaged 25-30 hours on-line each week, executing between 250 and 300 instructions per week.

His highest activity level occurred during the first two weeks of FEU when he executed over 500 instructions per week. During this period, an exercise simulating a crisis was conducted. During the exercise, crisis team members were logged on for long periods and some were using the system almost constantly. D was an active participant in the exercise.

Perhaps because D was already experienced when FEU began his error level was constant and low throughout FEU. He was already quite familiar with Sigma when these data were collected. His error rate never exceeded 2 errors per 100 instructions, and often it was under 1 per 100.

Table 18 shows D's Sigma instruction profile during FEU. Unlike User B, who was chief of a larger branch, D executed a wide variety of instructions. He had already established a pattern when FEU began. The core of his instruction use consisted of about 25 instructions, including "display/view message", "display file", "file/move", "delete message", "display text", "comment", "create message", "create text", "copy/put/movepickup text", "find string", and "alert". This branch chief reviewed his incoming messages and also created outgoing messages, using text objects as an aid to that task.

Late in the experiment D used some new instructions, such as "readdress message", "empty/sort file", and "highlight". Some of these had been added to Sigma during FEU at the request of the users. He found them helpful and began to use them regularly.

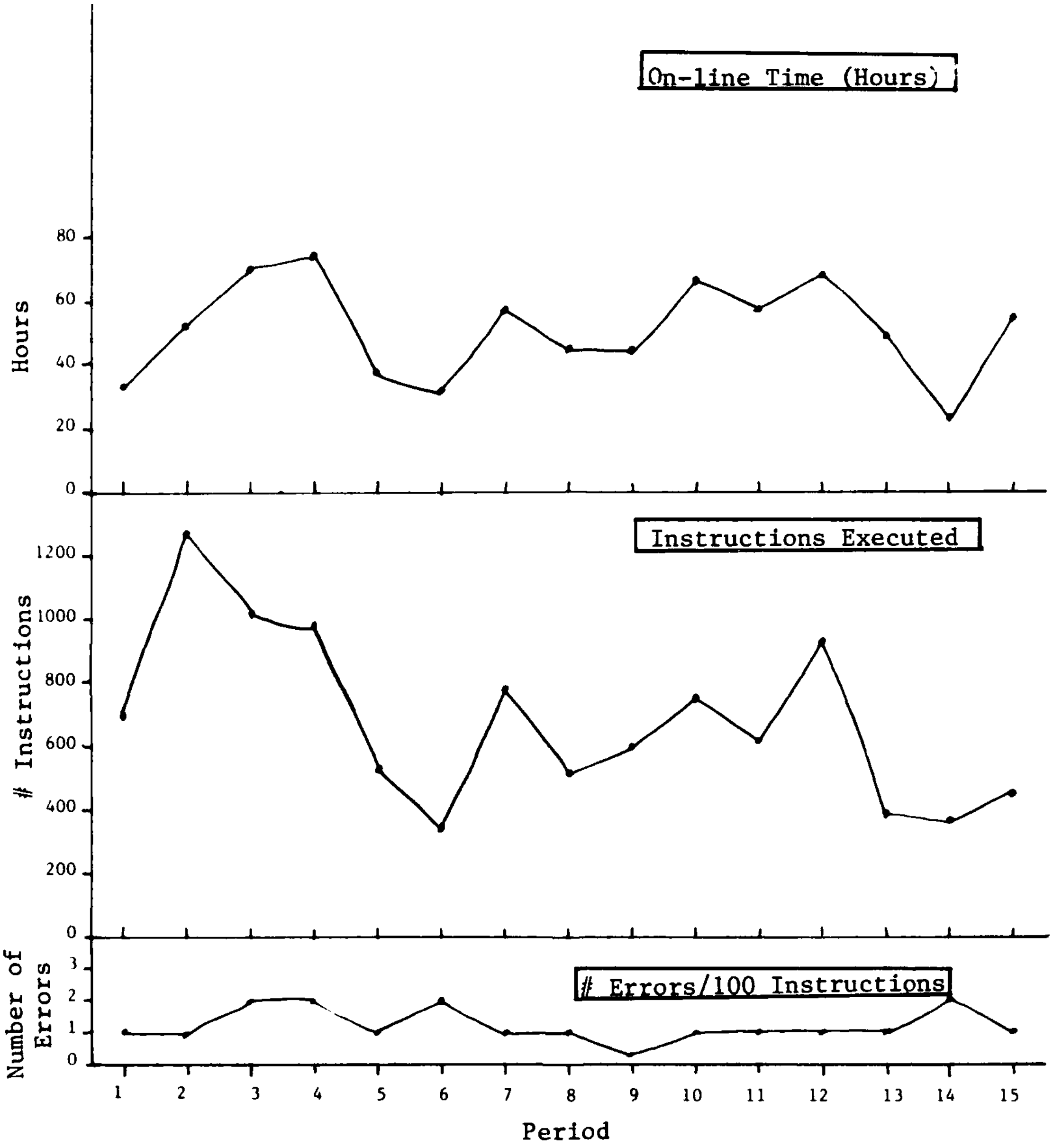


Figure 7. User D: Sigma Use

Table 18

User D: Instruction Use

Instruction	Period 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
display message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
display file	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
print	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view directory	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
find entry	x	x	x	x	x		x	x		x	x	x	x		
clear view windov	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
save/update	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
finish	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
file/move	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
delete message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
restrict/augment	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
backup	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
forward message	x	x	x	x	x	x	x				x		x		
display text	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
comment	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
coordinate	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
release	x	x	x	x			x	x	x		x		x		
create message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
create file	x	x	x	x		x	x				x				
create text	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
create selector	x	x	x	x		x	x	x		x	x	x	x		
view selector	x	x	x	x		x		x				x	x		x
copy/move...text	x	x	x	x	x		x	x	x	x	x	x	x	x	x
find string	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
alert	x	x	x	x	x	x	x	x	x		x	x	x		
route			x												
reply			x	x	x		x		x	x	x	x		x	
go to			x									x			
keyword message			x				x								
chop message				x											
copy message							x			x					
readdress							x				x	x			
empty/sort file							x	x	x	x	x	x		x	x
highlight								x	x			x			x

USER E

User E was an action officer in the branch responsible for staff action pertaining to offensive and defensive air operations. This branch should not be confused with the air desk officers in the Command Center who were watch standers monitoring current air activity. E began using Sigma during the exercise early in FEU. Figure 8 shows that, except for a few periods, his Sigma activity increased as FEU progressed, both in terms of on-line time and number of instructions executed. In the beginning, he was on-line less than 10 hours for two weeks and was executing between 170 and 240 instructions (he did not participate in the exercise). In period 12, he had a peak on-line time of almost 70 hours; his instruction use during this period had grown to 500 instructions. Periods 10-12 were periods of high system availability (see Figure 1); system up-time in periods 13 and 14 was lower, as was E's activity. Evidently he grew to like Sigma and began to rely on it more and more for daily message handling activities.

E's error rate was fairly low throughout FEU even though he had just begun using the system. It never exceeded 4 errors per 100 instructions and it was often less than 1 per 100.

Note in Table 19 that E's instruction repertoire was limited to a fairly small set of instructions throughout FEU. He used mostly "display message", "display file", "print", "file/move", "restrict/augment", and "delete message". His activity involved primarily incoming message review and file maintenance. He did expand his instruction use later in FEU to include "find entry" and the alert capability. He never took advantage of Sigma's ability to create outgoing messages.

USER F

User F was not in the Operations Directorate, the one selected for the experiment. He requested a Sigma account for his own message handling activities after having been a participant in the exercise held early in FEU. His branch was the Joint Petroleum Office, which revised, monitored, and coordinated matters pertaining to the supply of petroleum products within CINCPAC. His Sigma activity is documented in Figure 9 and Table 20.

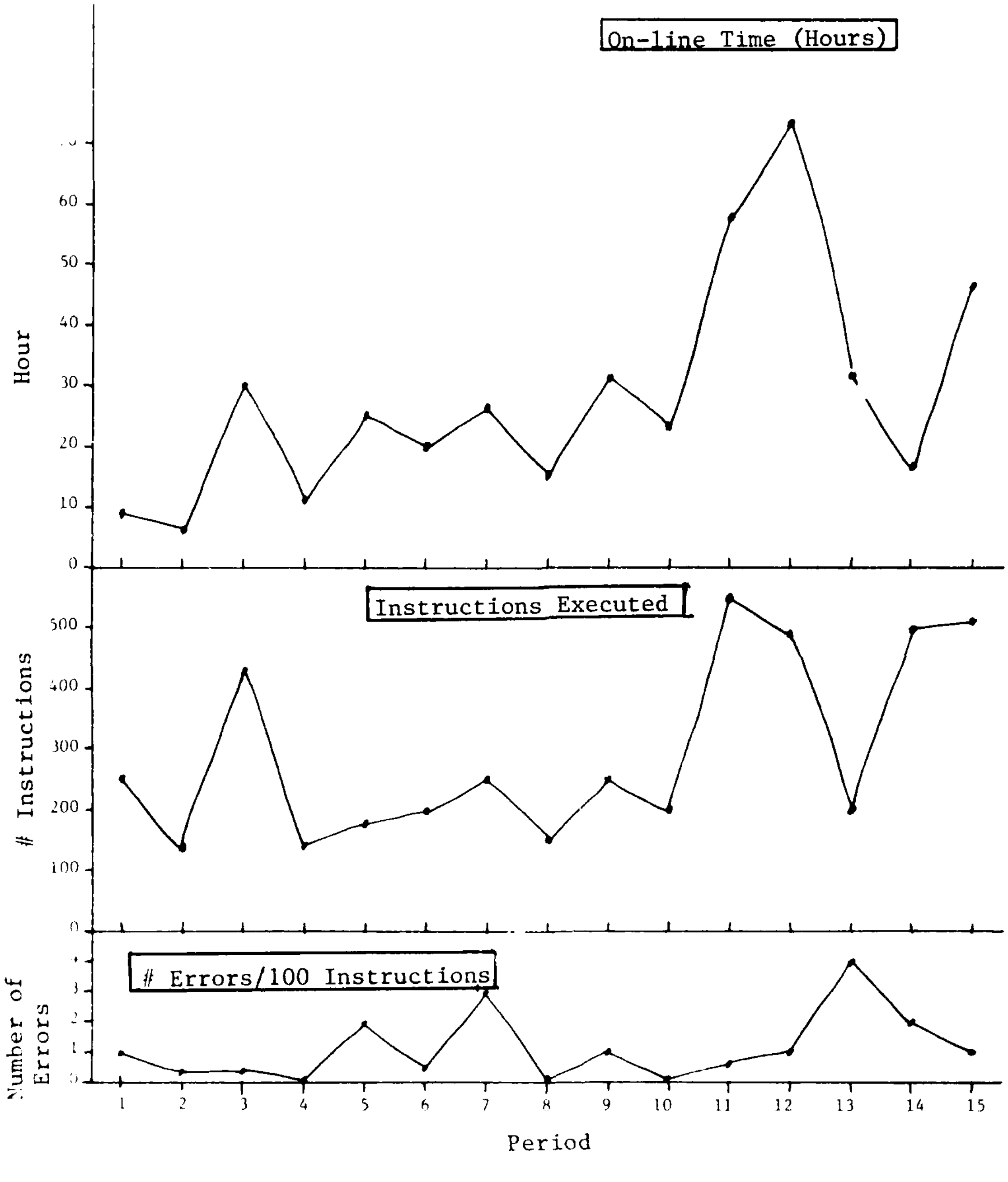


Figure 8. User E: Sigma Use

NON-SESSION TRANSCRIPT DATA PROCESSING PROGRAMS

Archive Retrieval Time Information Program

The process of measuring the age of a message when a request was made for its retrieval from the magnetic tape archive had two steps. First, an internal identification number (called the sequence number) was extracted from the DCF data block recorded when the display request was made. This was necessary because the originating time of the message was not itself recorded in that data block. This sequence number was used as an index into a log made when messages arrived from the AUTODIN network, where the DTG was recorded.

Acquiring these numbers required writing two programs. The first examined the DCF data. Each time a user attempted to display an archived message Sigma recorded the day and time (whether or not the user answered "yes" when asked if he wished to have the message retrieved from the archive), as well as the sequence number. The second program searched a source of data called the AUTODIN log, which was recorded on the tapes sent to MITRE every other week. In this log, a record was written whenever a message arrived at Sigma from the AUTODIN network. This record contained, among other things, a sequence number assigned to the message by Sigma, and the DTG of the incoming message, which established the origin time. In some cases this time preceded the arrival time of the message by as much as several days, if the message had been readdressed to an office in J3 by one of the original recipients. The age of the message, in days, was then computed by searching the AUTODIN log printout for a particular sequence number, and subtracting the date of origin of the message from the date of the retrieval inquiry. Results from this rogram were used in Section 8 of this report.

Data Collection Facility Programs

The results presented in Sections 9 and 10 were based on procedures previously written to reduce data captured by the Data Collection Facility (DCF). A brief description of these data reduction procedures was given in Reference 1.

Selector Counting Program

This program examined the preprocessed session transcript records. Whenever it found a typed "restrict" or "augment" instruction, it noted first whether the instruction used a typed selector or a previously stored and named selector; next whether or not the selection criterion for typed selectors was preceded by the logical operator "not"; then a user type, based on the user's job at CINCPAC; and finally the identification of the named selector or typed selector criterion that was being used. The program created a unique string out of these four components, and counted how many times each such created string occurred. There were about ten user types, and each user was assigned a type by someone well acquainted with the J3 organization. Results from this program were used in Section 5 of this report.

Restrict and Augment Level Counting

The purpose of this program was to find out how many successive "restrict" or "augment" instructions a user might carry out before reaching the subset of the message file that he wished to work with. To carry this out, the program first established a restrict/augment "task", and then kept track of the level of restricting or augmenting that took place during a task. The level count was a number which indicated the depth of restricting or augmenting within a task. Tasks were initiated by the first valid "restrict" or "augment" instruction within a session which changed the level count from zero to one, and were terminated when the level count next returned to zero. The level count was increased by one for each valid "restrict" or "augment" instruction (that is, one which was not followed by an error message). No distinction was made between instructions which included a typed selector and those which used a named and previously stored selector, even though these selectors might themselves have many logical components. The level count was reduced by one for each valid "backup one" instruction, and was set to zero by each "backup all", "display file", or end-of-session instruction. An analysis showed that changes associated with these instructions could be relied upon to produce the same effect as a "finish file" instruction, so this instruction was not separately monitored. Results from this program were used in Section 6 of this report.

Pattern Counting Program

The purpose of this program was to provide information on the frequency of occurrence of certain sequences of instructions. Because the number of possible instruction sequences was so large, no attempt was made to look for all combinations. Therefore, the program required that the first one or several instructions in the sequences of instructions it was looking for be fully specified in advance, and gave considerable freedom in describing these instructions (for example, in describing a typed instruction, one could look either for a specific second subfield or for any second subfield). One could specify up to 20 instructions in any one sequence. The program then tabulated all single instructions which immediately followed the defined sequence start.

Inputs to the program consisted of descriptions of starts of sequences. In each sequence start, each instruction was defined as a ten character string, similar to the last ten characters of the preprocessed session transcript records. The first character was either T or F, reflecting whether the instruction being described was a typed instruction or a function key instruction. If it was a function key instruction, the next three characters (the first subfield) contained a number from one to forty-two which designated a specific function key instruction, and the rest of the characters were left blank. If it was a typed instruction, the remaining nine characters contained values of instruction type and parameters that were to be looked for. If these subfields (one in the case of function key instructions, and three in the case of typed instructions) were filled with asterisks, then the program would recognize instructions which had any values at all in the corresponding subfields.

The program operated by examining successive sequences of session transcript records. If the start of the sequence matched a given description, then the program added one to the count associated with whatever instruction immediately followed. For specifications which included asterisks (indicating that all values were to be accepted for the fields marked with asterisks), the resulting count included all combinations encountered for the different values. Results from this program were used in Section 4 of this report.

In extracting these counts, some attempt was made to separate counts of those values in a secondary field for particular values of a preceding field. For example, in a typed instruction, if the first subfield showed it was either a "create" or a "reclassify" instruction, then counts were kept for the second subfield, which was used to record the security level of the item being created or reclassified. The results of this program are presented in Appendix A of this report.

Typed Instruction Counting Program

This program was used to get a more detailed breakdown on which typed instructions were used more frequently than others. Rather than simply note how many times a value occurred in the first subfield of a typed instruction, and then note the number of times various values occurred in the second subfield (as was done for the raw instruction counts, described later in this section), the program provided counts of all combinations of first and second subfields in typed instructions. It did this by treating each combination of first subfield and second subfield as a unique character string (using the hashing package), and then counting the number of occurrences of each such string. Results from this program were used in Sections 2 and 7 of this report.

Error Correlation Program

This program was used to provide some insight into the reasons for the error messages produced by Sigma. It looked through the preprocessed session transcript records until it encountered an error message, then backed up and looked at the instruction which preceded the error, and kept separate counts of these preceding instructions. Unique counts were kept for all function key instructions, and for all combinations of first and second words of typed instructions -- for example, separate counts were maintained for "display message" and "display file", which are two variations on the typed "display" instruction in which the second subfield differs. Results from this program were used in Section 3 of this report.

In particular, for typed "restrict" or "augment" instructions in which the selection criteria were spelled out as part of the instruction (rather than having previously been stored in a named selector) only information on the first criterion was kept. Counts were also kept of whether or not the first criterion was modified by the logical operator "not".

The Preprocessing Programs

The programs which did the session transcript preprocessing had four main components:

+ A package which extracted each raw session transcript line, converted it to EBCDIC, and passed it forward to individual processing routines.

+ A routine which filled out the first three fields of the output record (type, time, and function key/typed instruction/error message determination).

+ Two routines which further analyzed typed instructions, paying particular attention to those used to augment and restrict subsets of message files.

+ A support package for hashing character strings, so that they could be converted to numeric codes.

These routines, as well as the preprocessed session transcript data, will be stored in an orderly fashion at MITRE, should any need for further data analysis ever arise.

SESSION TRANSCRIPT DATA PROCESSING PROGRAMS

Raw Instruction Counting Program

This program simply counted the occurrences of particular values in the fields of the preprocessed session transcript records, without regard to the relationship of adjacent instructions (for example, an instruction was counted even if its successor happened to be an error message). The contents of the time field were ignored. These counts, which are presented in Appendix A, serve as a baseline for the counts presented in other sections whose results were derived from session transcript data.

period of Full Experimental Usage (FEU), from mid-February, 1979, to the end of September, 1979). The session transcript data shared each tape with other sorts of data, and were not necessarily recorded in chronological order. Finally, the entries varied widely in length and in syntactical form.

To overcome these deficiencies, a set of preprocessing programs was created. The purpose of these programs was to centralize these data on a more convenient medium (a demountable disk pack); to convert the data to EBCDIC once, rather than every time they were used; to rearrange them in strict chronological order; and to do some preliminary analysis so that all entries were stored in a compact, highly encoded fixed length format. This latter feature made it much easier for subsequent processing programs.

The final form of the data was 16-character records. Nearly 400,000 of them were written onto a disk pack, covering the period of FEU mentioned above. The fields in each record were:

1. A one-character field indicating record type (session start, identification, regular session line, or session end).

2. A five-character field containing time *information* -- month and day for session start records, and time of day for others.

3. A one-character field containing entry type in the regular session entries -- function key instruction, typed instruction, or error message.

4. A nine-character field with three three-character subfields containing syntactical information about typed instructions, many of which had multiple options in choice of functions and parameters.

In the process of converting the session transcript entries into fixed format records, some of the syntactical details of the typed instructions were dropped. Information about the first word (which specified the type of instruction, such as "display") and about the second word, which usually specified the item to which the action was being applied (such as "message" or "file"), was kept. When the third part of the instruction referred to a single word or phrase drawn from relatively small sets of words or phrases, that information was kept. However, when the instruction was relatively open-ended after the first two parts, some simplifying assumptions were made in order to keep the record length down and to get the parsing program running in a reasonable length of time.

many different instructions, some of which had many variations and options.

```
Sigma 2.34 Session started at Thursday, September 20, 1979 13:49:40-HST
Login by SMITH on TTY106 at SECRET

13:51:14: delete file pending
13:51:14: You cannot delete or restore your PENDING file.
13:52:02: display file pending
13:52:35: *clear view window*
13:52:56: display entry 2
13:55:27: log off

Sigma 2.34 Session started at Thursday, September 20, 1979 08:36:12-HST
Login by SMITH on TTY63 at SECRET

...
```

Figure 10. Sample Session Transcript

These session transcript lines were written into a special disk file assigned to each user. Once a week, this file was closed, and a new one started. Every two weeks, the two files for each user were copied to magnetic tape and deleted from the disk. The sequence of copying was often out of order -- that is, the second week's file usually appeared on the tape before the first's. The tapes were also used to record other data -- the DCF data mentioned earlier in this section, as well as information relating office and user names to the internal numbers used in the DCF data. There was also a file of AUTODIN log data -- records containing information about every incoming message. These tapes were sent to MITRE for further data reduction, and the DCF data extracted from them were the basis for previously published results (see Reference 1).

Preprocessing the Session Transcripts

In their original form, the session transcript data did not lend themselves readily to further analysis. They were written in ASCII code, and MITRE's principal computer facility used EBCDIC code. They occupied parts of many tapes (16 were used for the it had been examined and filled out by the Command Language Processor (CLP) -- that is, after it had been found to be a complete and valid Sigma instruction. Although Sigma allowed users much freedom in the use of abbreviations and in variations of instruction syntax, none of these abbreviations or variations show up in the session transcripts, because the CLP converted them all to standard form. In one sense, this was good -- it made the session transcript lines easier to process and analyze. However, much of the dialogue that took place between Sigma and its users was lost, because much of it dealt with changing what the user typed into a standard form understandable to the rest of Sigma. Only that standard form, and the system's responses to it, were recorded.

The Form of Session Transcripts

Most instructions in the session trancripts were followed by other instructions (occasionally interrupted by entries which marked the beginnings and ends of sessions). Often Sigma was unable to process an instruction, even though the instruction was perfectly valid. This may have been either because the context in which that instruction was used rendered it meaningless (such as asking to see the contents of the next message in a file when no file had yet been opened), or because the characteristics of the entities currently being dealt with (files, selectors, messages) meant that Sigma had nothing to do in order to satisfy the instruction (as in the case in which a user asked to see a list of all messages in the current file that had FLASH priority, and there were none). In either case, Sigma issued an "error" message, and the text of that message appeared as the next line in the session transcripts.

Figure 10 is an example of the transcript of a short session and the beginning of a second session. These transcripts consisted of one line of character data for each instruction entered and one line for each error message returned by the system. In addition, two lines of text containing information about the individual user, the office he worked in, the terminal being used, the date and time, and the security level of the session appeared at the beginning of each session. These two lines were separated from the previous session by a blank line, and another blank line separated them from the session they introduced. When a session was voluntarily terminated by a user, a special logoff line was written as the last line of the session. If the session was terminated involuntarily, as by a problem in Sigma or in the communication system connecting the user terminals to Sigma, this line did not appear. The lines themselves varied widely in length and content, since there were

SECTION 11

DATA ACQUISITION METHODS

INTRODUCTION

The results presented in the previous sections are based on data acquired from various sources. In this section, the data sources will be identified, along with a brief discussion of the methods used for extracting data from the sources and processing them to produce their final form. The discussions of the programs written to do this extraction and processing will be at a functional level, omitting most of the programming details.

Many of the discussions presented above used data extracted from the session transcripts. The first part of this section will be devoted to discussing those transcripts, and how data from them were preprocessed so that they could be more conveniently used by specific processing programs developed for each area of interest. The next part contains descriptions of the specific processing programs, and the final part contains descriptions of programs used to process data from sources other than the session transcripts.

SESSION TRANSCRIPTS

During the period of time that Sigma was being used by members of the Operations Directorate at Camp Smith, two types of data were automatically recorded -- Data Collection Facility (DCF) data, and session transcript data. DCF data consisted of binary blocks written by the internal subroutines of Sigma as it went through its operations in response to the user's instructions. These data were profuse and highly coded. In many cases, the sequence in which data blocks were written into the DCF did not strictly correspond to the time at which the events generating them were taking place. Furthermore, a major portion of the data recorded in each block was more descriptive of the internal operations of Sigma -- subroutine calls, serial numbers of messages and files, and the like -- than with the sequence of instructions a user was issuing.

The session transcripts, on the other hand, simply recorded the dialogue taking place between each user and Sigma. Each instruction a user issued to Sigma was written in character form to a special file assigned to that user. An instruction was recorded only after

On-line time was affected by three major factors. One was the user's job. Watch standers in the Command Center needed to be logged on almost constantly to monitor incoming message traffic, while most other users were logged on only for short periods of message handling activity. A second factor was terminal availability. In many CINCPAC offices there was only one terminal for four or five users, which limited the time any one user could spend on-line. The third factor was system availability. Obviously, when system availability was low, users could not spend as much time on-line.

Instruction use seemed to be more closely related to a user's job responsibilities than to the length of time he had been using the system. For the most part, users only executed those instructions which were obviously related to the message handling tasks that their job required. They rarely explored Sigma's other capabilities to see how they might be used to improve their message handling performance.

For most users tracked in this section, the level of error occurrence was low, about two per 100 instructions. In addition, it did not change much during FEU. This indicates that Sigma was a reasonably easy system to use, and also that users tended to stay with instructions they already knew how to use.

able to get his own Sigma account and settled into a fairly stable routine.

As Figure 9 demonstrates, once F received his own Sigma account (as opposed to using an account set up for the exercise), his activity level seemed to stabilize at somewhat lower figures (150-250 instructions, 3-10 hours per two week period) than during the exercise. This individual style reflected his day-to-day message handling activities and terminal availability; his office did not have a Sigma terminal so his on-line use was restricted to terminals in other offices. He might have used Sigma more had he had a terminal of his own.

F's error rate, shown in Figure 9, remained about the same throughout FEU at 3-5 errors per 100 instructions.

Table 20 illustrates F's instruction profile during FEU. From the beginning, it was varied. He used the basic message review functions ("display file", "display message", "restrict/augment"); since he received no incoming messages on Sigma other than comeback copies of messages he created (not being on J301's routing list), these functions were probably used to review datefiles, as well as his own personal files. In addition, he executed instructions associated with outgoing messages ("create message", "coordinate message", "release message", "copy/put/move/pickup text"). F found Sigma convenient for message creation. He also found the "delete message", "print", and "readdress" functions useful. This pattern of instruction use remained constant throughout FEU.

CONCLUSIONS

These studies do not verify early notions about the use of Sigma over a period of time. It was expected that instruction use would become diversified as users became more familiar with the message handling features that Sigma offered. This was not generally the case. Error levels had been expected to decrease when in fact they were low initially (1-2 errors per 100 instructions) and remained that way. The period of time involved in FEU (seven months) was somewhat short for this type of study; ideally, one should look at a stable, permanent system over a period of years. While it is risky to make generalizations about use of an automated system over time, there are some interesting features about the data.

Table 20

User F: Instruction Use

Instruction	Period 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
display message	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
display file	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
view message	x	x	-		x	x	x	x	x	x	x	x	x	x	x
print	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
view directory	x	x	-	x		x	x	x			x				x
find entry	x		-												
clear view window	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
save/update	x	x	-	x									x	x	x
finish	x	x	-	x	x		x	x		x					
file/move	x	x	-	x		x	x	x							
delete message	x	x	-	x		x	x	x	x	x	x	x	x	x	x
restrict/augment	x	x	-	x	x	x	x	x	x	x		x	x	x	x
backup	x	x	-	x											
release			-				x	x	x	x	x	x	x	x	x
reply	x		-												
display text	x		-												
comment	x	x	-	x											
coordinate	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
create message	x	x	-	x	x	x	x	x	x	x	x	x	x	x	x
create file	x	x	-	x		x	x								
keyword message			-												x
create text	x		-												
create selector	x	x	-	x		x									
view selector	x		-												
copy/move...text	x	x	-	x			x	x	x	x	x	x	x	x	x
alert	x		-							x			x		
copy entry			-											x	
go to		x	-	x											
readdress		x	-	x		x	x	x	x			x	x	x	

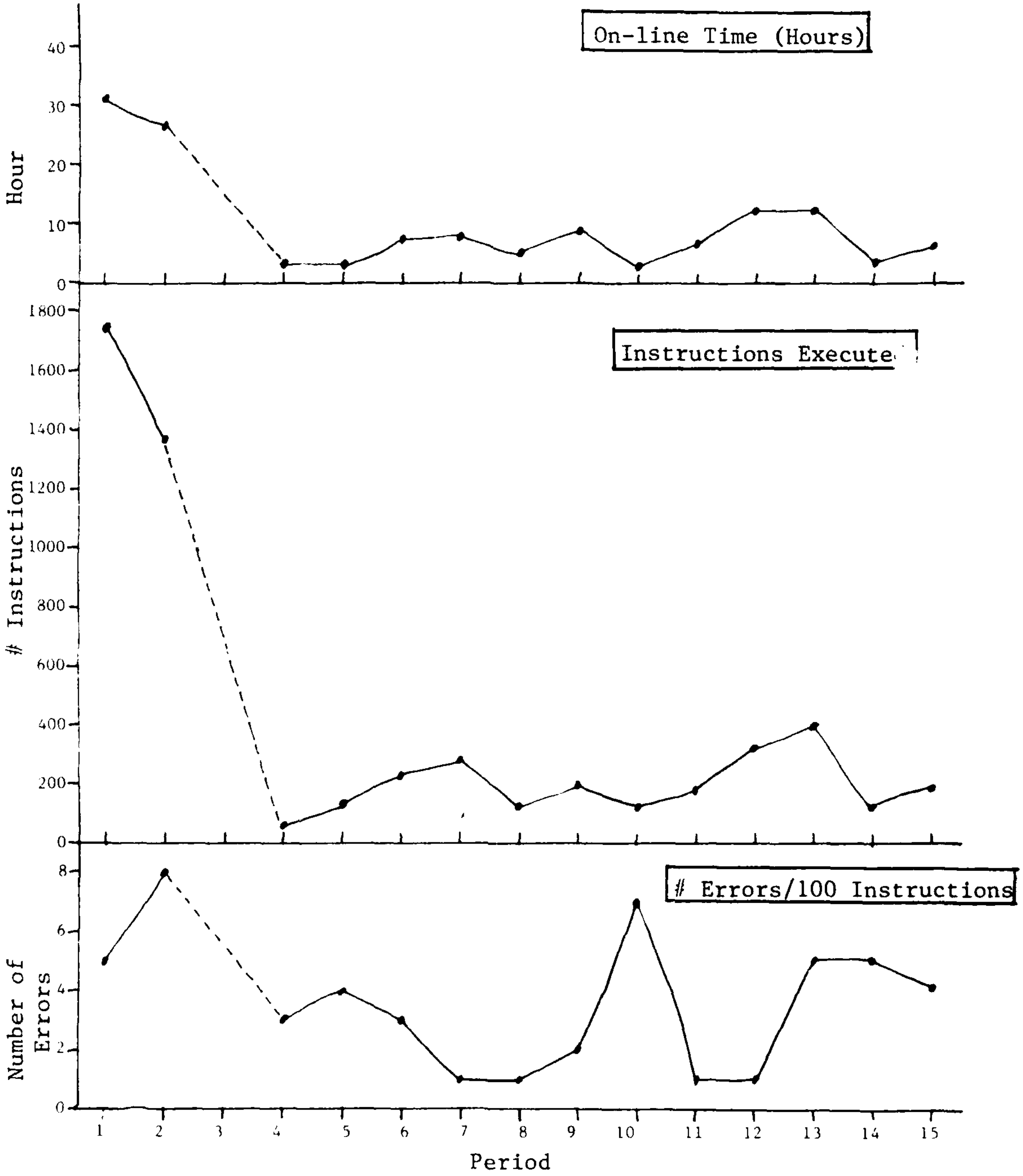


Figure 9. User F: Sigma Use

Table 19

User E: Instruction Use

	Period														
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
display message	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
display file	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
view message			x		x	x	x	x	x	x					
print	x		x	x	x	x	x	x	x	x	x	x	x		x
view directory	x	x	x											x	
clear view window	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
save/update											x				
finish	x	x		x	x	x	x	x	x	x	x	x	x	x	x
file/move	x	x	x	x											
delete message	x	x	x	x		x	x	x	x	x	x	x	x	x	x
restrict/augment	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
backup		x	x	x		x			x	x	x	x	x	x	x
create file	x		x												
coordinate					x										
readdress					x										
find entry									x	x	x	x	x	x	x
alert									x	x			x		x
chop message												x			

LIST OF REFERENCES

1. N. C. Goodwin and S W. Hosmer, "A User-Oriented Evaluation of Computer Aided Message Handling," MTR 3920, Bedford, Mass.: The MITRE Corp., April 1980.

APPENDIX A

RAW SESSION TRANSCRIPT DATA COUNTS

There were 398,694 records in the preprocessed session transcript data for the period of Full Experimental Usage (from 22 February 1979 to 29 September 1979). In this appendix, some raw statistics about those records are presented. These statistics can be used as a baseline for comparing the data presented in earlier sections of this report. It should be remembered that these counts are gross; that is, no attempt was made to invalidate records because of their relative position with respect to other records (such as instructions which were followed by error messages), and very few relationships between components of individual records were taken into account. For an explanation of what the records consist of, and how these statistics were extracted from them, refer to Section 11.

At the highest level, the 398,694 records were broken down as follows: 9,021 of them were session start records; 9,021 were logon identification records; 7,382 were normal termination (or logoff) records; 1,639 were abnormal termination records (added by the preprocessing programs, since they didn't exist in the original data); and 371,631 were regular records (records which occurred during a session, and which included instructions as well as error responses).

The 9,021 logon records were distributed among 171 users for whom records were kept; the names will not be reported here. Of the 9,021 logons, 756 were at the unclassified level, 244 at the confidential level, 6,786 at the secret level, and 1,077 at the top secret level (although AUTODIN, the military's digital information network, supplied Sigma no top secret messages).

The 371,631 regular records consisted of 224,217 function key instructions, 19,000 error messages, and 128,414 typed instructions. Table A-1 gives counts of those function key instructions which were used more than 500 times.

Table A-1

Frequently Used Function Key Instructions
(including those which caused errors)

Instruction	Uses
delete file entry	39,957
delete next entry	37,876
display file entry	21,168
clear view window	19,165
finish displayed object	16,640
show open file	16,024
view next entry	11,343
display next entry	10,517
view file entry	9,458
backup one	7,124
print displayed object	6,672
print file entry	5,312
alert on/off	4,847
save displayed object	3,022
update displayed object	2,208
view file directory	1,582
copy text	1,441
release open message	1,281
coordinate open message	1,095
chop no open message	941
view text directory	891
readdress entry	658
show open text object	614
move text	579

The 19,000 errors generated 132 different error messages. Table A-2 shows those that occurred more than 200 times (with abbreviated message contents).

Table A-2

Frequently Occurring Error Messages
(including both "user" and "system" errors)

Error Message	Occurrences
Your selector returned no entries	4094
The requested file is being updated	1664
You have no current entry	1464
Typed instruction input prompting	977
You do not have anything viewed	812
You will receive a citation when the message is retrieved	806
The requested string was not found	753
You are currently at the end of the file	560
You do not have an open object	437
The requested message is being updated	428
The requested message is not being retrieved	387
Restrict file display not implemented yet	339
Cannot locate specified file	299
The help facility is being activated	291
The requested message does not exist	283
You cannot access this message	279
Your entry list matched nothing displayed	276
You do not have an open file	273
There is no open file ...	264

The 128,414 typed instructions made use of thirty-five different first words (functions). Table A-3 shows those that occurred more than 100 times.

Table A-3

First Words in Typed Instructions

Word	Occurrences
display	45,140
move	18,553
restrict	18,436
route	14,128
delete	10,223
find	4,662
create	2,835
file	2,225
reply	2,104
view	1,524
identify	1,097
forward	1,077
restore	929
comment	873
empty	698
get	654
system	633
sort	599
abort	363
print	287
put	222
augment	212
lesson	212
quit	205
action	177
keyword	105

In most (but not all) typed instructions, the second word used modified the first. Table A-4 gives the frequency of use of all second words (or phrases) which appeared more than 100 times.

Table A-4

Second Words in Typed Instructions

Word or Phrase	Occurrences
file	43,324
entry	40,806
text	8,768
datefile	9,894
(none needed)	4,208
string	1,999
message	1,123
selector	798
bottom	641
status	554
top	368
version	226
open file	151
news	112
file directory	100

In the 18,648 instructions in which the first word was either "restrict" or "augment", 5,925 instructions used previously named and stored selectors, while 12,723 instructions had a selector supplied as part of the typed instruction. Of these 12,723 typed selectors, 805 were modified by the logical operator "not".

Finally, for those 2,848 instructions which either created a Sigma entity (file, message, text object, or selector) or reclassified a text object, 1,354 specified no security classification for the result, 79 specified unclassified, 227 specified confidential, 1138 specified secret, and 50 specified top secret. In those cases where no classification was specified, Sigma assigned the logon level classification to the resulting entity.

GLOSSARY

AUTODIN	Automatic Digital Network
CINCPAC	Commander in Chief, Pacific
DCF	Data Collection Facility
DDO	Duty Director of Operations
DTG	date-time group
FBIS	Foreign Broadcast Information Service
FEU	Full Experimental Use
J3	CINCPAC Operations Directorate
LDMX	Local Digital Message Exchange
LEU	Limited Experimental Use
MME	Military Message Experiment

DISTRIBUTION LIST

INTERNAL

D-11

J. J. Croke

D-40

B. M. Horowitz
F. R. Murphy
J. Mitchell

D-50

F. Chess
E. Lobenstein

D-70

W. S. Attridge
E. H. Bensley
D. I. Buckley
E. L. Burke
J. B. Campbell
C. J. Carter
J. A. Clapp
E. S. Herndon
E. L. Lafferty
W. H. Melahn
A. D. McKersie
D. A. McQueen
K. C. Rowe
J. F. Spitzer
K. M. Wadman

D-71

S. L. Smith

D-72

R. J. Anderson
T. K. Backman
J. N. Boolukos
M. Corasick

D-72 (contd.)

A. S. Cressey
N. C. Goodwin (20)
J. P. Hogan

D-73

W. Amory
T. L. Connors
C. Engelman
G. D. Knapp
D. G. Miller (20)
S. L. Rawls
L. C. Record
R. E. Wagner
J. J. Wandling

D-75

S. R. Ames
M. Gasser
D. E. Kane
C. W. McClure
R. S. Popp
P. S. Tasker

DISTRIBUTION LIST (Continued)

EXTERNAL

Office of the Under Secretary of Defense for Research & Engineering
Communications Command Control and Intelligence
Pentagon
Washington, DC 20301

Lt. Col. Wilcox

Joint Chiefs of Staff
Pentagon
Washington, DC 20301

Operations Directorate -- Information Systems Division

Defense Communications Agency
8th & South Courthouse Road
Alexandria, VA 22204

Attn: Code J34

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

Information Processing Techniques Office (5)

Chief of Naval Operations
Department of the Navy
Washington, DC 20350

Attn: OP94

Commander-in-Chief Pacific
Camp Smith
Oahu, HI 96860

Attn: J3

Headquarters
U.S. Air Force
Pentagon
Washington, DC 20330

Attn: AT-/XOKCR

DISTRIBUTION LIST (Continued)

WWMCCS System Engineering Organization
Defense Communications Agency
8th & South Courthouse Road
Alexandria, VA 22204

Col. Pixton
Maj. Noll

Command and Control Technical Center
Pentagon
Washington, DC 20301

Attn: Code C634

Rome Air Development Center
Griffiss Air Force Base
Rome, NY 13441

D. K. Plante

Air Force Intelligence Service
Fort Belvoir, VA 22060

M. W. Lamour

Naval Intelligence Processing Systems Support Activity
Hoffman Building
2461 Eisenhower Avenue
Alexandria, VA 22331

NIC OOQ4 -- Mr. Gray

Defense Communications Engineering Center
Derey Engineering Building
1860 Wieble Avenue
Reston, VA 22090

Attn: Code 3800

Office of Naval Research
Ballston Centre Tower #1
800 North Quincy Street
Arlington, VA 22217

Attn: Code 437

DISTRIBUTION LIST (Continued)

Headquarters
Department of the Army
Forrestal Building
Washington, DC 20314

Attn: DAAC-SIF

Naval Ocean Systems Center
San Diego, CA 92152

Attn: Code 8122

Naval Sea Systems Command
Washington, DC 20362

Attn: Code 047

Naval Research Laboratory
Washington, DC 20375

Attn: Code 7503 (5)

Defense Intelligence Agency
Pentagon
Washington, DC 20301

Attn: R50-34 RSP-2

Headquarters
TRADOC
Fort Monroe
Hampton, VA 23651

Attn: ATCD-C-C

Commander
7th Signal Command
Fort Ritchie, MD 21719

Attn: CCCN-PO-N

U.S. Readiness Command
MacDill Air Force Base, FL 33608

Attn: RCJ6-T

DISTRIBUTION LIST (Continued)

Commander
U.S. Army Communications Command
Fort Huachuca, AZ 85613

Attn: CC-OPA-WA
Attn: CC-OPS-T

Headquarters
Strategic Air Command
Offutt Air Force Base, NE 68005

Attn: DXT

ARPA PAC
Camp Smith
Oahu, HI 96860

ARPA Headquarters
EUCOM
APO NY 09128

Supreme Headquarters Allied Powers Europe
Pentagon
Washington, DC 20301

W. Stoney

National Science Foundation
1800 G Street, N.W.
Washington, DC 20240

P. Custer

National Bureau of Standards
Gaithersburg, MD 20760

Dr. S. Kimbelton

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90291

K. Uncapher (2)

DISTRIBUTION LIST (Continued)

CTEC
7777 Leesburg Pike
Falls Church, VA 22043

Dr. E. Bersoff (3)

Commander
Naval Electronic Systems Command
Washington, DC 20360

Attn: Code 310

Headquarters Pacific Air Force/DOCO
Hickam AFB
Hawaii 96853

Maj. R. D. Gordon

Planning Research Center
7600 Old Spring House Road
McLean, VA 22102

Bernard Rudwick

Chief of Naval Research
Office of Naval Research (ONR-100)
Bellston Centre Tower #1
800 North Quincy Street
Arlington, VA 22217

Air Force Communications Computer Programming Center
Analysis & Design
Tinker AFB, OK 73145

Maj. R. Harris

Naval Telecommunications Command
3801 Nebraska Avenue N.W.
Washington, DC 20390

A. Drain

Personnel and Training Activity Office
Bldg. 200, Rm. 3422
Washington, DC 20390

T. Young

DISTRIBUTION LIST (Concluded)

College of Business Administration
University of Hawaii
2404 Maile Way
Honolulu, HI 96822

R. Panko

Ford Aerospace & Communications Corporation
Western Development Laboratories
Division
3939 Fabian Way
Palo Alto, CA 94303

R. T. Gilbert MS L-40

BJR Inc.
3174 Porter Drive
Palo Alto, CA 94025

J. Blair

Xerox Corporation (A1-47)
701 S. Aviation Boulevard
El Segundo, CA 90245

W. Verplank

Bolt, Beranek & Newman
50 Moulton Street
Cambridge, MA 02138

C. Mooers
T. Myers
R. Pew